

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
«Πλατφόρμα Υποβολής, Διαχείρισης και Αξιολόγησης
Φοιτητικών Εργασιών»

Εργασίες Μαθήματος

<p>Νέα Εργασία 1 Νέα Εργασία 1 Υποβολή: 2025-05-11 14:17:00 έως 2025-05-11 19:18:00</p> <p>Υποβολή Εργασίας</p>
<p>Εργασία 2 Περιγραφή Εργασία2 Υποβολή: 2025-05-14 19:18:00 έως 2025-05-20 19:18:00</p> <p>Η υποβολή δεν έχει ξεκινήσει ακόμα.</p>
<p>Εργασία 0 Εργασία 0 Υποβολή: 2025-04-08 15:19:00 έως 2025-05-05 15:19:00</p> <p>Η προθεσμία υποβολής έχει λήξει.</p>
<p>Επίλυση Ασκήσεων Παραγώγων Βιβλίο Σελ 44-46 - Ασκήσεις 1-20 Υποβολή: 2025-05-15 20:00:00 έως 2025-05-22 23:55:00</p> <p>Η υποβολή δεν έχει ξεκινήσει ακόμα.</p>

Φοιτητής

ΘΩΜΑΣ-ΡΑΦΑΗΛ ΜΠΑΚΑΡΟΣ
134001

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Ιούνιος 2025

Πλατφόρμα υποβολής, διαχείρισης και αξιολόγησης φοιτητικών εργασιών

Κωδικός: 25150

Φοιτητής: Μπάκαρος Θωμάς-Ραφαήλ

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 08-03-2025

Ημερομηνία περάτωσης Π.Ε. 18-05-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή **Μπάκαρου Θωμά-Ραφαήλ** που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Περίληψη

Η παρούσα εργασία αφορά την ανάπτυξη μιας διαδικτυακής πλατφόρμας για την υποβολή, διαχείριση και αξιολόγηση φοιτητικών εργασιών. Στόχος του συστήματος είναι η διευκόλυνση της εκπαιδευτικής διαδικασίας μέσω της ψηφιακής διαχείρισης των εργασιών, προσφέροντας ένα λειτουργικό και φιλικό περιβάλλον για φοιτητές, καθηγητές και διαχειριστές. Η πλατφόρμα επιτρέπει στους φοιτητές την υποβολή εργασιών και την παρακολούθηση της προόδου τους, ενώ οι καθηγητές μπορούν να οργανώνουν μαθήματα, να αναθέτουν εργασίες, να αξιολογούν και να παρέχουν σχόλια. Για την υλοποίηση χρησιμοποιήθηκαν οι τεχνολογίες PHP, το framework Laravel και η βάση δεδομένων MySQL.

« Platform for submitting, managing and evaluating student work »

Abstract

This thesis focuses on the development of a web-based platform for the submission, management, and evaluation of student assignments. The aim of the system is to facilitate the educational process through the digital management of assignments, offering a functional and user-friendly environment for students, teachers, and administrators. The platform allows students to submit their assignments and monitor their progress, while teachers can organize courses, assign tasks, evaluate submissions, and provide feedback. For the implementation of the platform, the technologies PHP, the Laravel framework, and the MySQL database were utilized.

Ευχαριστίες

Θέλω να ευχαριστήσω τους γονείς μου, τους φίλους μου και τον επιβλέπων κ. Τσιακμάκη Κυριάκο για τη βοήθεια και τη κατανόηση τους.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	viii
Κεφάλαιο 1ο: Εισαγωγή	9
1.1 Εισαγωγή	9
1.2 Η δομή του κειμένου	11
Κεφάλαιο 2ο: Online Συστήματα Διαχείρισης Εργασιών	13
2.1 Moodle	13
2.2 Open edX	14
2.3 Google Classroom	16
Κεφάλαιο 3ο: Γλώσσες και Περιβάλλοντα προγραμματισμού	19
3.1 Η γλώσσα PHP	19
3.2 Το Framework Laravel	22
3.3 MySQL	24
Κεφάλαιο 4ο: Η Διαδικτυακή Πλατφόρμα Διαχείρισης Εργασιών	29
4.1 Εισαγωγή – Η διαδικασία	29
4.2 Η διαδικασία	30
4.3 Διαδικτυακή εφαρμογή - ιστοσελίδα	38
4.4 Η Βάση μας	51
4.5 Ασφάλεια συστήματος	53
Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης	56
ΒΙΒΛΙΟΓΡΑΦΙΑ	58
ΠΑΡΑΡΤΗΜΑ Α	59

Κατάλογος Σχημάτων

Εικόνα 2.1: Moodle – ρύθμιση για υποβολή εργασίας	13
Εικόνα 2.2: Open edX – ρύθμιση για υποβολή εργασίας	15
Εικόνα 2.3: Google Classroom– ρύθμιση για υποβολή εργασίας	17
Εικόνα 3.1: Key Advantages of PHP Development.....	21
Εικόνα 4.1: Διάγραμμα Διαδικασίας Υποβολής Εργασίας (Student Workflow)	30
Εικόνα 4.2: Διάγραμμα Διαδικασίας Βαθμολόγησης Εργασιών (Teacher Workflow)	32
Εικόνα 4.3: Διάγραμμα Διαχείρισης Μαθημάτων (Admin/Teacher).....	33
Εικόνα 4.4: Διάγραμμα Διαδικασίας Εγγραφής Νέου Φοιτητή/Καθηγητή	34
Εικόνα 4.5: Διάγραμμα Διαδικασίας Ελέγχου Deadlines και Πρόσβασης Υποβολής (Ασφάλεια Υποβολής).....	35
Εικόνα 4.6: Η πρώτη σελίδα.....	38
Εικόνα 4.7: Σύνδεση	39
Εικόνα 4.8: Για Εγγραφή	40
Εικόνα 4.9: Αρχική - Admin	40
Εικόνα 4.10: Διαχείριση Μαθημάτων - Προσθήκη Νέου Μαθήματος 1 - Admin	41
Εικόνα 4.11: Διαχείριση Μαθημάτων - Προσθήκη Νέου Μαθήματος 2 - Admin	42
Εικόνα 4.12: Αρχική - Teacher	42
Εικόνα 4.13: Τα Μαθήματά μου- Teacher	43
Εικόνα 4.14: Διαχείριση Φοιτητών - Φοιτητές Μαθήματος - Teacher	43
Εικόνα 4.15: Διαχείριση Εργασιών - Εργασίες Μαθήματος - Teacher	44
Εικόνα 4.16: Νέα Εργασία - Teacher	44
Εικόνα 4.17: Μετά την προσθήκη Νέας Εργασίας - Teacher.....	45
Εικόνα 4.18: Επεξεργασία Εργασίας και ρύθμιση Deadlines - Teacher	45
Εικόνα 4.19: Υποβολές αρχείων εργασιών των φοιτητών - Teacher.....	46
Εικόνα 4.20: Υποβολές εργασιών φοιτητών με μη βαθμολογημένη εργασία- Teacher	46
Εικόνα 4.21: Βαθμολογημένες Υποβολές εργασιών φοιτητών - Teacher	47
Εικόνα 4.22: Αρχική σελίδα για τον φοιτητή – Student.....	47
Εικόνα 4.23: Σελίδα με τα μαθήματα που συμμετέχει ο φοιτητής- Student	47
Εικόνα 4.24: Εργασίες για το Μάθημα Μαθηματικά – όπως το βλέπει ο φοιτητής- Εργασίες εντός και εκτός προθεσμίας - Student.....	48
Εικόνα 4.25: Σελίδα για υποβολή αρχείου - Student.....	48
Εικόνα 4.26: Επιτυχή υποβολή αρχείου - Student.....	49
Εικόνα 4.27: Σελίδα με τις εργασίες – φαίνεται ότι έγινε η υποβολή - Student	49
Εικόνα 4.28: Σελίδα με τις βαθμολογίες ανά εργασία – χωρίς βαθμό - Student	50
Εικόνα 4.29: Σελίδα με τις βαθμολογίες ανά εργασία – με βαθμό - Student	50
Εικόνα 4.24: Οι πίνακες της βάσης μας	51

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Στην εποχή της μεγάλης τεχνολογικής ανάπτυξης οι σύγχρονες εκπαιδευτικές διαδικασίες στοχεύουν στη χρήση πληροφοριακών συστημάτων και διαδικτυακών πλατφορμών. Οι παραδοσιακές μέθοδοι υποβολής φοιτητικών εργασιών που στηρίζονταν στην έντυπη μορφή και στη φυσική παράδοση στον διδάσκοντα έχουν πλέον αντικατασταθεί σε μεγάλο βαθμό από ηλεκτρονικές διαδικασίες. Η ανάγκη για εξοικονόμηση χρόνου, η μείωση του διοικητικού φόρτου και η διευκόλυνση της επικοινωνίας μεταξύ φοιτητών και καθηγητών οδήγησαν στην υλοποίηση πλατφορμών υποβολής, διαχείρισης και αξιολόγησης εργασιών.

Οι πλατφόρμες αυτές επιτρέπουν τη συγκέντρωση, οργάνωση και διαχείριση όλων των σχετικών δεδομένων με τις εργασίες των φοιτητών, παρέχοντας στους διδάσκοντες εύκολη πρόσβαση στα υποβληθέντα αρχεία, δυνατότητες άμεσης αξιολόγησης και παροχής σχολίων. Από την άλλη πλευρά, οι φοιτητές αποκτούν τη δυνατότητα να παρακολουθούν την πορεία των εργασιών τους, να υποβάλλουν τις εργασίες τους σε προκαθορισμένα χρονικά πλαίσια και να ενημερώνονται άμεσα για τις βαθμολογίες και τα σχόλια που λαμβάνουν.

Η χρήση τέτοιων πλατφορμών δεν περιορίζεται μόνο στην αποτελεσματική διαχείριση του εκπαιδευτικού έργου αλλά συμβάλλει καθοριστικά στη βελτίωση της ποιότητας της εκπαιδευτικής διαδικασίας στην καλύτερη οργάνωση των μαθημάτων και στη διασφάλιση της διαφάνειας κατά την αξιολόγηση.

Παρά την ύπαρξη πολλών έτοιμων λύσεων στην αγορά, όπως το Moodle, το Open edX και το Google Classroom, προέκυψε η ανάγκη για την ανάπτυξη μιας πιο εξειδικευμένης πλατφόρμας που να καλύπτει συγκεκριμένες ανάγκες του ακαδημαϊκού στο οποίο απευθύνεται. Τα υπάρχοντα συστήματα είτε κρίθηκαν πολύπλοκα και δύσκολα στη χρήση είτε γιατί δεν παρείχαν την απαραίτητη ευελιξία για την εξατομικευμένη προσαρμογή στις ιδιαίτερες ανάγκες των χρηστών.

Η έλλειψη ενός εύχρηστου και φιλικού και αποτελεσματικού εργαλείου διαχείρισης εργασιών οδήγησε στην ανάπτυξη μιας νέας πλατφόρμας προσαρμοσμένης στις ανάγκες τόσο των φοιτητών όσο και των διδασκόντων. Στόχος ήταν η απλοποίηση της διαδικασίας υποβολής και αξιολόγησης των φοιτητικών εργασιών, η άμεση πληροφόρηση των χρηστών και η δημιουργία ενός συστήματος με εύχρηστο περιβάλλον εργασίας.

Η πλατφόρμα που υλοποιήθηκε αποτελεί ένα ολοκληρωμένο διαδικτυακό σύστημα υποβολής, διαχείρισης και αξιολόγησης φοιτητικών εργασιών. Οι βασικοί ρόλοι που υποστηρίζονται είναι ο φοιτητής, ο καθηγητής και ο διαχειριστής του συστήματος.

Οι φοιτητές έχουν τη δυνατότητα να εγγράφονται σε μαθήματα, να βλέπουν τις διαθέσιμες εργασίες κάθε μαθήματος, να υποβάλλουν τις εργασίες τους ηλεκτρονικά εντός προκαθορισμένων χρονικών περιθωρίων και να ενημερώνονται για τη βαθμολογία και τα σχόλια που λαμβάνουν. Παράλληλα να μπορούν να παρακολουθούν την πορεία των υποβολών τους και τις σχετικές ημερομηνίες προθεσμιών.

Οι καθηγητές έχουν πρόσβαση σε όλα τα μαθήματα που τους έχουν ανατεθεί μπορούν να δημιουργούν και να τροποποιούν και να διαγράφουν εργασίες, να καθορίζουν προθεσμίες υποβολής και να διαχειρίζονται τις υποβολές των φοιτητών. Παρέχεται η δυνατότητα βαθμολόγησης, παροχής σχολίων και αναλυτικής παρακολούθησης της προόδου των φοιτητών ανά εργασία και μάθημα.

Ο διαχειριστής του συστήματος έχει πλήρη έλεγχο επί όλων των δεδομένων με δυνατότητα δημιουργίας, επεξεργασίας και διαγραφής μαθημάτων και χρηστών και ανάθεσης μαθημάτων σε καθηγητές.

Η πλατφόρμα αυτή απλοποιεί σημαντικά την εκπαιδευτική διαδικασία και παρέχει ένα εργαλείο εύχρηστο και πρακτικό τόσο για φοιτητές όσο και για καθηγητές. Η ηλεκτρονική διαχείριση των εργασιών μειώνει το διοικητικό βάρος, βελτιώνει την οργάνωση του εκπαιδευτικού έργου και συμβάλλει στην καλύτερη παρακολούθηση της προόδου των φοιτητών.

Μέσω της άμεσης πρόσβασης στα αποτελέσματα και τα σχόλια των διδασκόντων, οι φοιτητές έχουν την ευκαιρία να βελτιώσουν τις δεξιότητές τους, να εντοπίσουν τις αδυναμίες τους και να εστιάσουν σε τομείς που απαιτούν περαιτέρω βελτίωση. Η διαφάνεια στη διαδικασία αξιολόγησης ενισχύει το αίσθημα δικαιοσύνης και αντικειμενικότητας.

Η πλατφόρμα υλοποιήθηκε με τη χρήση των παρακάτω τεχνολογιών:

- **PHP:** Η βασική γλώσσα προγραμματισμού για την ανάπτυξη του backend της εφαρμογής, επιλεγμένη για την ευελιξία, τη σταθερότητα και την εκτενή υποστήριξή της από την παγκόσμια κοινότητα.
- **Laravel Framework:** Το κύριο πλαίσιο εργασίας για την οργάνωση της εφαρμογής. Παρέιχε τη δυνατότητα γρήγορης ανάπτυξης, καθαρής αρχιτεκτονικής και ενσωμάτωσης σύγχρονων τεχνικών ασφαλείας.
- **MySQL:** Το σύστημα διαχείρισης βάσεων δεδομένων, το οποίο χρησιμοποιήθηκε για την ασφαλή και αξιόπιστη αποθήκευση όλων των δεδομένων του συστήματος.
- **HTML5, CSS3 & Bootstrap:** Για την ανάπτυξη ενός φιλικού και responsive περιβάλλοντος χρήστη, που εξασφαλίζει ομαλή λειτουργία σε διαφορετικές συσκευές και αναλύσεις οθονών.
- **Blade Templating Engine:** Για την οργάνωση των views και τη δημιουργία δυναμικού περιεχομένου με απλό και κατανοητό τρόπο.

Η υλοποίηση της παρούσας πλατφόρμας προσέφερε σημαντικά οφέλη, τόσο σε τεχνικό όσο και σε προσωπικό επίπεδο. Η ανάπτυξη ενός πλήρους πληροφοριακού συστήματος συνέβαλε στην γνώση και στη κατανόηση των τεχνολογιών PHP, Laravel και MySQL σε επίπεδο θεωρίας όσο και σε πρακτικές υλοποίησης.

Η ολοκλήρωση του έργου ανέδειξε την ικανότητα σχεδιασμού και υλοποίησης μιας εφαρμογής με πραγματικές λειτουργικές απαιτήσεις για τις δεξιότητες στην ανάλυση προβλημάτων, στον καθορισμό απαιτήσεων και στον σχεδιασμό της βάσης δεδομένων και στην ανάπτυξη φιλικού περιβάλλοντος διεπαφής.

Η διαδικασία υλοποίησης του έργου ενίσχυσε την ικανότητα διαχείρισης χρόνου, την οργάνωση πολύπλοκων λειτουργικών απαιτήσεων και την επίλυση πρακτικών προβλημάτων που προκύπτουν κατά την ανάπτυξη ενός συστήματος.

1.2 Η δομή του κειμένου

Η παρούσα εργασία είναι οργανωμένη σε πέντε βασικά κεφάλαια, καθένα από τα οποία εξετάζει και αναλύει διαφορετικές πτυχές της μελέτης και της υλοποίησης του συστήματος διαχείρισης φοιτητικών εργασιών.

Στο Κεφάλαιο 1: Εισαγωγή, γίνεται μια γενική παρουσίαση της θεματολογίας και της σημασίας των διαδικτυακών πλατφορμών διαχείρισης φοιτητικών εργασιών. Ακολουθεί η ανάλυση της ανάγκης για την υλοποίηση ενός τέτοιου συστήματος, η περιγραφή της πλατφόρμας που αναπτύχθηκε, καθώς και η χρησιμότητά της στην εκπαιδευτική διαδικασία. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των τεχνολογιών που χρησιμοποιήθηκαν και το προσωπικό όφελος που προέκυψε από την ανάπτυξη του έργου.

Το Κεφάλαιο 2: Online Συστήματα Διαχείρισης Εργασιών παρουσιάζει και αναλύει τρία σημαντικά και ευρέως χρησιμοποιούμενα συστήματα: το Moodle, το Open edX και το Google Classroom. Στόχος του κεφαλαίου είναι να γίνει μια συγκριτική παρουσίαση των δυνατοτήτων αυτών των συστημάτων σε σχέση με τη λειτουργικότητα της παρούσας υλοποίησης.

Στο Κεφάλαιο 3: Γλώσσες και Περιβάλλοντα Προγραμματισμού, περιγράφονται οι βασικές τεχνολογίες που αξιοποιήθηκαν για την ανάπτυξη της πλατφόρμας. Παρουσιάζονται αναλυτικά η γλώσσα προγραμματισμού PHP, το σύγχρονο framework Laravel που χρησιμοποιήθηκε για την ανάπτυξη του backend, καθώς και η MySQL ως σύστημα διαχείρισης βάσεων δεδομένων. Παράλληλα, αναφέρονται τα πλεονεκτήματα και οι λόγοι επιλογής αυτών των τεχνολογιών.

Το Κεφάλαιο 4: Η Διαδικτυακή Πλατφόρμα Διαχείρισης Εργασιών εστιάζει στην αναλυτική παρουσίαση της διαδικασίας υλοποίησης του συστήματος. Περιγράφονται οι κύριες διαδικασίες που υποστηρίζει η πλατφόρμα, η σχεδίαση της διαδικτυακής εφαρμογής και της βάσης δεδομένων, καθώς και τα ζητήματα ασφάλειας που αντιμετωπίστηκαν κατά την υλοποίηση. Το κεφάλαιο συνοδεύεται από διαγράμματα ροής, απεικονίσεις διεπαφών και τεχνικές λεπτομέρειες υλοποίησης.

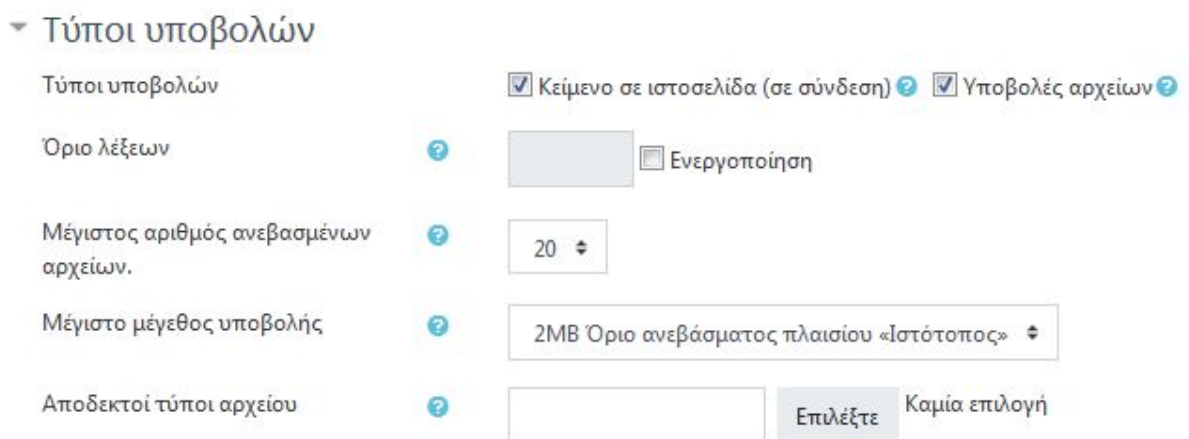
Το Κεφάλαιο 5: Συμπεράσματα και Προτάσεις Βελτίωσης, γίνεται μια συνολική αποτίμηση της εργασίας. Παρουσιάζονται τα συμπεράσματα που προέκυψαν από την ανάπτυξη της πλατφόρμας, αξιολογείται η αποτελεσματικότητά της και διατυπώνονται προτάσεις για μελλοντικές επεκτάσεις και βελτιώσεις.

Η εργασία ολοκληρώνεται με τη Βιβλιογραφία, όπου καταγράφονται όλες οι πηγές που χρησιμοποιήθηκαν, και το Παράρτημα Α, το οποίο περιλαμβάνει επιπλέον υλικό και τεχνικές λεπτομέρειες που κρίθηκαν χρήσιμες για την πληρέστερη κατανόηση του έργου.

Κεφάλαιο 2ο: Online Συστήματα Διαχείρισης Εργασιών

2.1 Moodle

Το Moodle αποτελεί μία από τις πιο διαδεδομένες πλατφόρμες διαχείρισης εκπαιδευτικού περιεχομένου (Learning Management System – LMS) παγκοσμίως και χρησιμοποιείται ευρέως σε ακαδημαϊκά ιδρύματα για την υποστήριξη της εξ αποστάσεως και συνδυαστικής εκπαίδευσης. Ως λογισμικό ανοιχτού κώδικα, παρέχει ένα ευρύ φάσμα εργαλείων για τη δημιουργία, διαχείριση και αξιολόγηση μαθημάτων και φοιτητικών εργασιών. [1]



Τύποι υποβολών

Τύποι υποβολών Κείμενο σε ιστοσελίδα (σε σύνδεση) Υποβολές αρχείων

Όριο λέξεων Ενεργοποίηση

Μέγιστος αριθμός ανεβασμένων αρχείων

Μέγιστο μέγεθος υποβολής

Αποδεκτοί τύποι αρχείου Καμία επιλογή

Εικόνα 2.1: Moodle – ρύθμιση για υποβολή εργασίας

[https://seminars.etwinning.gr/pluginfile.php/95071/mod_assign/intro/moodle124.png]

Δυνατότητες Σχετικές με τη Διαχείριση Εργασιών

1. Υποβολή Εργασιών (Assignments)

Στο Moodle, οι διδάσκοντες έχουν τη δυνατότητα να δημιουργούν δραστηριότητες τύπου "Assignment" για την υποβολή φοιτητικών εργασιών. Οι φοιτητές μπορούν να ανεβάζουν τα αρχεία τους απευθείας στην πλατφόρμα, ενώ οι εκπαιδευτικοί έχουν άμεση πρόσβαση στις υποβολές μέσω της ίδιας διεπαφής.

2. Ρύθμιση Προθεσμιών και Περιορισμών

Οι διδάσκοντες μπορούν να ορίσουν ακριβείς ημερομηνίες έναρξης και λήξης υποβολών, καθώς και χρονικά περιθώρια για καθυστερημένες παραδόσεις με αντίστοιχη επίπτωση στη βαθμολογία. Το σύστημα παρέχει ειδοποιήσεις στους φοιτητές για τις επικείμενες προθεσμίες.

3. Βαθμολόγηση και Σχόλια

Η πλατφόρμα υποστηρίζει διαφορετικές μεθόδους βαθμολόγησης (μονοψήφιες ή πολυψήφιες κλίμακες, ποσοστά, περιγραφική αξιολόγηση κ.ά.) και δίνει τη δυνατότητα καταγραφής αναλυτικών σχολίων πάνω σε κάθε εργασία. Επίσης υποστηρίζεται η ανώνυμη αξιολόγηση για αντικειμενικότερη βαθμολόγηση.

4. Ανατροφοδότηση και Επαναξιολόγηση

Το Moodle δίνει τη δυνατότητα στους φοιτητές να δουν τη βαθμολογία και τα σχόλια που έχουν λάβει για κάθε εργασία ενώ οι καθηγητές μπορούν να επιτρέψουν ή να περιορίσουν την υποβολή διορθωμένων εκδόσεων μιας εργασίας, ανάλογα με την πολιτική του μαθήματος.

5. Αναλυτικές Αναφορές

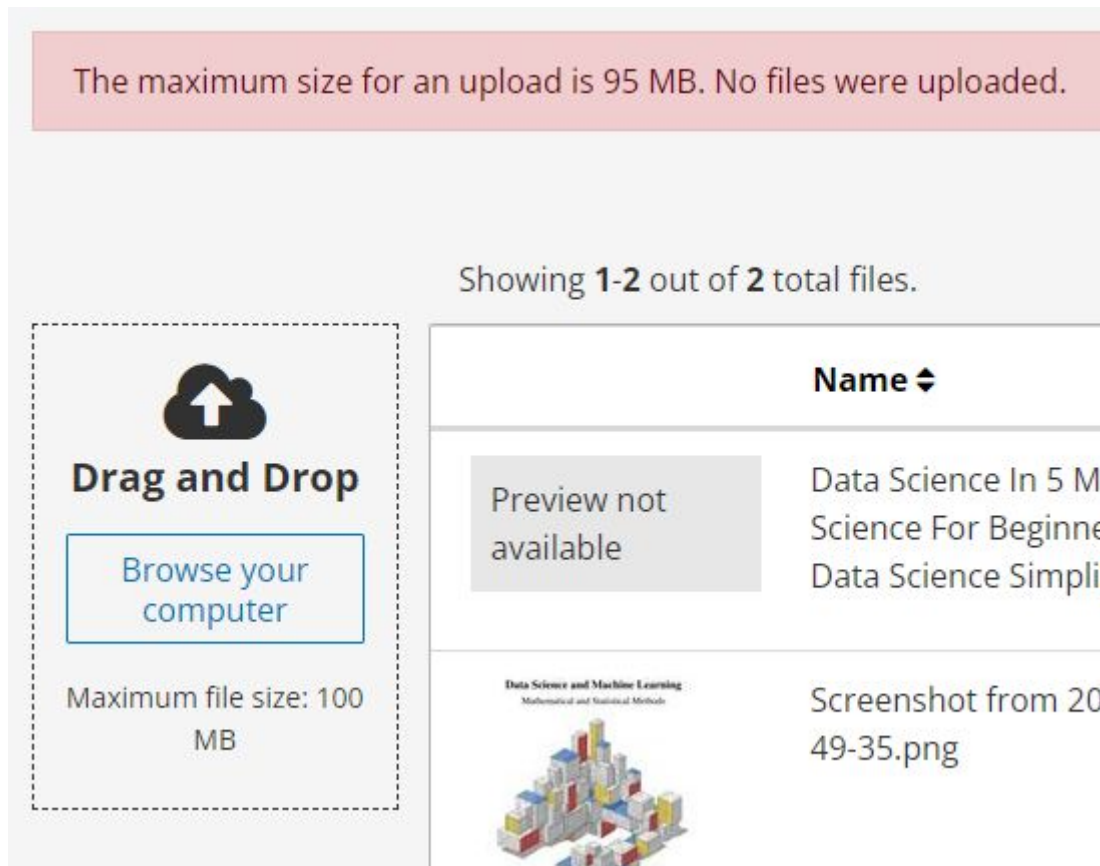
Το σύστημα έχει εργαλεία παρακολούθησης και εξαγωγής στατιστικών σχετικά με την πρόοδο των φοιτητών, τις υποβολές εργασιών και την απόδοση της τάξης διευκολύνοντας την αξιολόγηση τόσο σε ατομικό όσο και σε συνολικό επίπεδο.

Το Moodle αποτελεί ένα ολοκληρωμένο και ευέλικτο εργαλείο για τη διαχείριση και αξιολόγηση φοιτητικών εργασιών, υποστηρίζοντας πληθώρα επιλογών που προσαρμόζονται εύκολα στις ανάγκες του κάθε εκπαιδευτικού οργανισμού. Η κύρια αδυναμία του εντοπίζεται στη συχνά περίπλοκη παραμετροποίηση και στη λιγότερο φιλική προς το χρήστη διεπαφή σε σύγκριση με πιο μοντέρνες πλατφόρμες, γεγονός που ενδέχεται να δημιουργήσει δυσκολίες σε φοιτητές και καθηγητές με περιορισμένες τεχνολογικές δεξιότητες.

2.2 Open edX

Το Open edX αποτελεί μία από τις πιο σύγχρονες και ισχυρές πλατφόρμες ανοιχτού κώδικα για τη διαχείριση διαδικτυακής εκπαίδευσης και την παροχή εκπαιδευτικού περιεχομένου σε μεγάλη κλίμακα. Δημιουργήθηκε αρχικά από το MIT και το Harvard με στόχο την υποστήριξη μαζικών ανοιχτών διαδικτυακών μαθημάτων (MOOCs) ωστόσο η αρχιτεκτονική του έχει εξελιχθεί σε τέτοιο βαθμό ώστε να μπορεί να υποστηρίξει τόσο μαζικά όσο και εξειδικευμένα εκπαιδευτικά περιβάλλοντα. Η

πλατφόρμα έχει υιοθετηθεί από κορυφαία ακαδημαϊκά ιδρύματα και οργανισμούς παγκοσμίως, παρέχοντας ισχυρά εργαλεία για την υποβολή, αξιολόγηση και διαχείριση φοιτητικών εργασιών. [2]



Εικόνα 2.2: Open edX – ρύθμιση για υποβολή εργασίας

[<https://us1.discourse-cdn.com/flex020/uploads/openedx/original/2X/b/b830b55be59233ec6e98ef181af085acc1da5132.png>]

Το σύστημα προσφέρει έναν πλήρως οργανωμένο μηχανισμό για τη διαχείριση εκπαιδευτικών μαθημάτων και δραστηριοτήτων, επιτρέποντας την εύκολη δημιουργία εκπαιδευτικού περιεχομένου, την παρακολούθηση της πορείας των φοιτητών και την αξιολόγηση των δεξιοτήτων τους μέσω μιας πληθώρας εργαλείων και μεθόδων. Η διαδικασία υποβολής φοιτητικών εργασιών είναι πλήρως ενσωματωμένη στο περιβάλλον της πλατφόρμας επιτρέποντας τόσο τη διαχείριση κειμενικών εργασιών όσο και την υποβολή αρχείων σε διαφορετικές μορφές. Οι εκπαιδευτές μπορούν να δημιουργούν ειδικές δραστηριότητες τύπου “Open Response Assessment”, όπου οι φοιτητές υποβάλλουν εκτενείς απαντήσεις ή αρχεία και στη συνέχεια αξιολογούνται από τους καθηγητές ή ακόμα και μέσω peer-assessment μηχανισμών για την αξιοποίηση της συλλογικής αξιολόγησης για εκπαιδευτικούς σκοπούς.

Η διαχείριση των προθεσμιών υποβολής είναι ιδιαίτερα ευέλικτη στο Open edX καθώς παρέχει τη δυνατότητα καθορισμού συγκεκριμένων χρονικών παραθύρων για την υποβολή εργασιών με πλήρη έλεγχο σχετικά με το αν επιτρέπεται η καθυστερημένη υποβολή και υπό ποιες προϋποθέσεις. Δίνεται

η δυνατότητα καθορισμού διαφορετικών σταδίων αξιολόγησης για τους εκπαιδευτές να καθορίσουν αν οι εργασίες θα αξιολογηθούν αυτόματα, θα περάσουν από χειροκίνητη αξιολόγηση ή θα υποστούν διαδικασία peer review.

Ένα πλεονέκτημα της πλατφόρμας αποτελεί το ισχυρό σύστημα παρακολούθησης της πορείας των φοιτητών και οι αναλυτικές αναφορές προόδου. Οι εκπαιδευτές μπορούν να παρακολουθούν όχι μόνο τις υποβολές των εργασιών αλλά και τον χρόνο συμμετοχής των φοιτητών, τις δραστηριότητες στις οποίες έχουν λάβει μέρος και τα ποσοστά επιτυχίας στις διάφορες δοκιμασίες και αξιολογήσεις. Αυτό το επίπεδο ανάλυσης επιτρέπει την πιο στοχευμένη υποστήριξη φοιτητών που ενδέχεται να αντιμετωπίζουν δυσκολίες και τη συνολική αξιολόγηση της αποτελεσματικότητας του εκπαιδευτικού περιεχομένου.

Η ασφάλεια των δεδομένων αποτελεί βασικό πυλώνα του Open edX. Οι υποβολές των εργασιών αποθηκεύονται με ασφαλείς μεθόδους και υποστηρίζεται πλήρως η κρυπτογράφηση των προσωπικών δεδομένων των χρηστών. Το σύστημα υποστηρίζει μηχανισμούς ειδοποιήσεων μέσω των οποίων τόσο οι φοιτητές όσο και οι εκπαιδευτές ενημερώνονται για κρίσιμες προθεσμίες, νέες εργασίες και σημαντικές ανακοινώσεις.

Παρά τα σημαντικά πλεονεκτήματα που προσφέρει το Open edX παρουσιάζει και ορισμένες αδυναμίες, ιδιαίτερα σε ό,τι αφορά την πολυπλοκότητα της εγκατάστασης και της παραμετροποίησής του. Απαιτείται σημαντική τεχνική κατάρτιση για την ορθή εγκατάσταση του συστήματος και τη διαχείριση των υποδομών ενώ η διαμόρφωση της διεπαφής και των ρυθμίσεων συχνά προϋποθέτει εξειδικευμένες γνώσεις. Παρόλα αυτά η ισχυρή του αρχιτεκτονική και οι δυνατότητες επεκτασιμότητας το καθιστούν ιδανικό για μεγάλης κλίμακας εκπαιδευτικά ιδρύματα που επιθυμούν ένα πλήρως παραμετροποιήσιμο και επαγγελματικού επιπέδου σύστημα διαχείρισης εκπαιδευτικού περιεχομένου και αξιολόγησης.

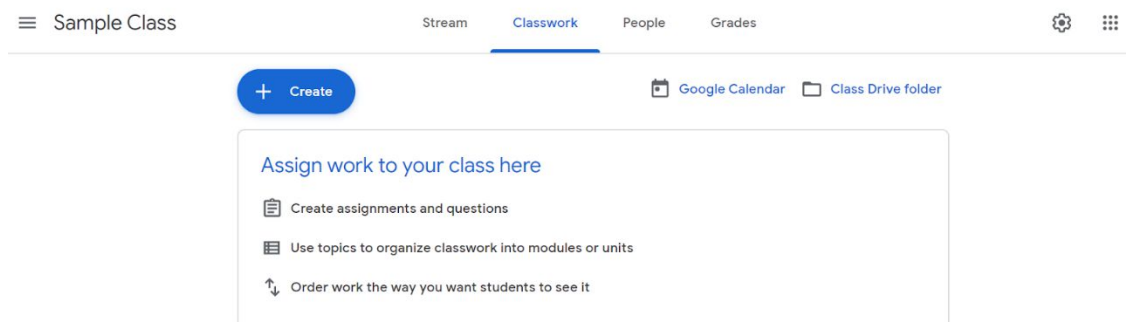
Το Open edX αποτελεί μια ιδιαίτερα ισχυρή και ολοκληρωμένη λύση για την υποστήριξη της διαχείρισης φοιτητικών εργασιών και της αξιολόγησης της εκπαιδευτικής δραστηριότητας. Παρά τις αυξημένες απαιτήσεις σε επίπεδο τεχνικής υποστήριξης, οι δυνατότητες που προσφέρει τόσο στους φοιτητές όσο και στους εκπαιδευτές είναι εξαιρετικά προηγμένες και καλύπτουν πλήρως τις ανάγκες ενός σύγχρονου εκπαιδευτικού οργανισμού.

2.3 Google Classroom

Το Google Classroom αποτελεί μια ελαφριά αλλά εξαιρετικά λειτουργική πλατφόρμα διαχείρισης ψηφιακής τάξης, η οποία έχει αναπτυχθεί από την Google με σκοπό να υποστηρίξει την εκπαιδευτική

διαδικασία τόσο σε πρωτοβάθμιο και δευτεροβάθμιο επίπεδο όσο και σε περιβάλλοντα τριτοβάθμιας εκπαίδευσης. Η απλότητα και η ευχρηστία και η άμεση ενσωμάτωση με άλλες υπηρεσίες της Google καθιστούν το Classroom ιδιαίτερα δημοφιλές εργαλείο σε σχολεία και πανεπιστήμια που επιθυμούν μία γρήγορη και ευέλικτη λύση για την ηλεκτρονική υποστήριξη των μαθημάτων. [3]

Στο πλαίσιο της διαχείρισης και αξιολόγησης φοιτητικών εργασιών, το Google Classroom προσφέρει έναν εξαιρετικά απλό και διαισθητικό μηχανισμό για τη δημιουργία και ανάθεση εργασιών από τον καθηγητή προς τους φοιτητές. Ο διδάσκων μπορεί να δημιουργήσει μία εργασία επιλέγοντας τον τίτλο, τις οδηγίες, τις σχετικές ημερομηνίες και τον τρόπο υποβολής. Οι φοιτητές λαμβάνουν ειδοποίηση μέσω email ή μέσω της ίδιας της πλατφόρμας και μπορούν να επισυνάψουν αρχεία είτε από το τοπικό τους σύστημα είτε απευθείας από το Google Drive, παρέχοντας έτσι ευελιξία στην υποβολή. Το σύστημα υποστηρίζει έλεγχο προθεσμιών με κάθε εργασία να συνοδεύεται από συγκεκριμένη ημερομηνία λήξης μετά την οποία οι υποβολές καταγράφονται ως εκπρόθεσμες.



Εικόνα 2.3: Google Classroom– ρύθμιση για υποβολή εργασίας

[https://lh6.googleusercontent.com/lz-K3gnMM1agAFBhFkSritbTc9ZE6Mwpub1ef6ewYr2k0YF8BRx9M19S9trL0ZOfCxKXgGvy8FVEpLQmY-gUJ75NHfQXijgpytAJJ_BWzS-8MEPIHcLmPo4OED_zdh-2aNxUcdL]

Η διαδικασία αξιολόγησης των εργασιών είναι εξίσου απλοποιημένη. Ο καθηγητής μπορεί να προβεί σε βαθμολόγηση απευθείας μέσα από την πλατφόρμα, να προσθέσει σχόλια είτε γενικά είτε επάνω στο ίδιο το αρχείο του φοιτητή (μέσω του Google Docs) και να επιστρέψει την εργασία στον φοιτητή με σχετική ανατροφοδότηση. Η επανυποβολή είναι δυνατή εφόσον το επιτρέπει ο διδάσκων γεγονός που υποστηρίζει μια πιο συνεργατική και εξελικτική προσέγγιση στην εκπαιδευτική διαδικασία.

Ένα ιδιαίτερο πλεονέκτημα του Google Classroom είναι η απρόσκοπτη διασύνδεση με το οικοσύστημα εργαλείων της Google, όπως Google Docs, Sheets, Slides και Drive. Αυτό επιτρέπει την εύκολη συγγραφή, αποθήκευση, διαμοιρασμό και αξιολόγηση εργασιών χωρίς την ανάγκη εξωτερικών εργαλείων ή υπηρεσιών. Το Classroom προσφέρει ένα απλό σύστημα αρχείων και ιστορικού, το οποίο βοηθά στην παρακολούθηση των εργασιών που έχουν υποβληθεί, επιστραφεί ή παραμείνει σε εκκρεμότητα.

Η απλότητα του Google Classroom έχει και ορισμένους περιορισμούς. Δεν προσφέρει το εύρος λειτουργιών και παραμετροποίησης που βρίσκουμε σε πιο εξειδικευμένα LMS, όπως το Moodle ή το Open edX. Δεν έχει εκτεταμένες αναφορές ή εργαλεία ανάλυσης δεδομένων απόδοσης και απουσιάζει η δυνατότητα ρύθμισης πιο σύνθετων τύπων αξιολόγησης ή σταδίων υποβολής. Εξαρτάται σε μεγάλο βαθμό από τη χρήση προσωπικού ή ιδρυματικού Google λογαριασμού, γεγονός που μπορεί να δημιουργήσει εμπόδια σε περιβάλλοντα όπου δεν είναι δυνατή η χρήση Google Workspace.

Το Google Classroom είναι μία από τις πιο προσιτές και άμεσα χρησιμοποιήσιμες λύσεις για την ηλεκτρονική διαχείριση φοιτητικών εργασιών. Αν και δεν προορίζεται για πολύπλοκες ή εξειδικευμένες εκπαιδευτικές ανάγκες, καλύπτει με ευκολία τις βασικές απαιτήσεις ενός σύγχρονου ψηφιακού μαθήματος, επιτρέποντας γρήγορη ανάθεση, υποβολή και αξιολόγηση εργασιών σε περιβάλλοντα που δίνουν έμφαση στην απλότητα και την ευχρηστία.

Κεφάλαιο 3ο: Γλώσσες και Περιβάλλοντα προγραμματισμού

Για την ανάπτυξη της διαδικτυακής πλατφόρμας διαχείρισης φοιτητικών εργασιών επιλέχθηκαν τεχνολογίες οι οποίες υποστηρίζουν γρήγορη ανάπτυξη και καλή διαχείριση και ασφαλή αποθήκευση δεδομένων. Οι βασικές τεχνολογίες που αξιοποιήθηκαν είναι η PHP ως βασική γλώσσα προγραμματισμού στο backend, το Laravel ως σύγχρονο framework που οργανώνει και επεκτείνει τις δυνατότητες της PHP και η MySQL ως σύστημα διαχείρισης βάσης δεδομένων.

Η PHP αποτέλεσε τη θεμελιώδη γλώσσα ανάπτυξης της εφαρμογής λόγω της ενσωμάτωσής της στο Laravel και της ευρείας υποστήριξής της από web servers. Το Laravel, βασισμένο στην PHP έχει την ταχύτερη και πιο οργανωμένη ανάπτυξη της εφαρμογής και έχει δυνατότητες όπως διαχείριση routes, session, template rendering μέσω Blade, αλλά και έλεγχο πρόσβασης. Παρά ότι στην παρούσα υλοποίηση έγινε αποφυγή των Eloquent Models για απλοποίηση χρησιμοποιήθηκαν πλήρως οι δομές του MVC που προσφέρει το Laravel.

Η MySQL χρησιμοποιήθηκε για την αποθήκευση όλων των δεδομένων της πλατφόρμας όπως χρήστες, μαθήματα, εργασίες, υποβολές και βαθμολογίες. Μέσω των κατάλληλων ερωτημάτων SQL με χρήση του DB Facade, εξασφαλίστηκε άμεση και αξιόπιστη επικοινωνία της εφαρμογής με τη βάση χωρίς την ανάγκη χρήσης ORM.

3.1 Η γλώσσα PHP

Η PHP (Hypertext Preprocessor) αποτελεί μία από τις πιο δημοφιλείς και διαδεδομένες γλώσσες προγραμματισμού στον χώρο της ανάπτυξης διαδικτυακών εφαρμογών. Δημιουργήθηκε το 1994 από τον Rasmus Lerdorf και έκτοτε εξελίχθηκε σε μια πλήρως δυναμική και αντικειμενοστραφή γλώσσα, η οποία υποστηρίζει την ανάπτυξη τόσο απλών ιστοσελίδων όσο και σύνθετων διαδικτυακών συστημάτων υψηλής κλίμακας. [4-7] Το μεγαλύτερο πλεονέκτημά της είναι η άμεση ενσωμάτωσή της σε web περιβάλλοντα καθώς εκτελείται στον server και παράγει δυναμικό περιεχόμενο HTML που αποστέλλεται στον client.

Η PHP έχει σχεδιαστεί με βασική σκέψη την ευκολία μάθησης και χρήσης και παράλληλα προσφέρει υψηλές δυνατότητες προσαρμογής υποστηρίζοντας ένα μεγάλο εύρος βιβλιοθηκών και πλαισίων εργασίας (frameworks) που διευκολύνουν την ανάπτυξη οργανωμένων και ασφαλών εφαρμογών. Στις νεότερες εκδόσεις της η γλώσσα υποστηρίζει πλήρως αντικειμενοστραφή προγραμματισμό, διαχείριση εξαιρέσεων, αυστηρούς τύπους δεδομένων και σύγχρονες πρακτικές ανάπτυξης λογισμικού, ενισχύοντας την αξιοπιστία και την επεκτασιμότητά της.

Στην παρούσα εργασία η PHP αποτέλεσε τη βασική γλώσσα για την υλοποίηση της λογικής της διαδικτυακής πλατφόρμας διαχείρισης φοιτητικών εργασιών. Όλα τα backend μέρη της εφαρμογής, όπως η διαχείριση χρηστών, η διασύνδεση με τη βάση δεδομένων, η επεξεργασία αιτημάτων HTTP και η διαχείριση των εκπαιδευτικών ροών υλοποιήθηκαν με χρήση PHP. Ο ρόλος της γλώσσας αυτής είναι καθοριστικός και αναλαμβάνει την επεξεργασία όλων των δεδομένων που εισάγονται ή ανακτώνται από τη βάση, την εκτέλεση των επιχειρησιακών κανόνων και την παραγωγή δυναμικού περιεχομένου που προβάλλεται στον τελικό χρήστη.

Η σύνδεση της PHP με τη βάση δεδομένων επιτυγχάνεται μέσω του μηχανισμού PDO ή της χρήσης έτοιμων εργαλείων που παρέχουν τα frameworks, όπως το DB Facade στο Laravel. Αυτή η δυνατότητα μας δίνει την άμεση εκτέλεση SQL ερωτημάτων, την ασφαλή διαχείριση δεδομένων και την προστασία απέναντι σε κοινές απειλές όπως οι επιθέσεις SQL Injection.

Η επιλογή της PHP για την ανάπτυξη του συστήματος βασίστηκε κυρίως στην σταθερότητα της γλώσσας και στην ευρεία κοινότητα υποστήριξης που διαθέτει και στη μεγάλη διαθεσιμότητα εργαλείων και βιβλιοθηκών που μειώνουν το χρόνο ανάπτυξης. Η PHP είναι πλήρως συμβατή με όλα τα σύγχρονα συστήματα φιλοξενίας ιστοσελίδων κάτι που διευκολύνει την εγκατάσταση και λειτουργία της πλατφόρμας σε διαφορετικά περιβάλλοντα παραγωγής.

Η PHP παραμένει μέχρι και σήμερα μία από τις κυρίαρχες γλώσσες στον χώρο του διαδικτύου, εξυπηρετώντας εκατομμύρια εφαρμογές και ιστοσελίδες σε όλο τον κόσμο. Στην παρούσα εργασία η χρήση της αποδείχθηκε κρίσιμη για την επιτυχή υλοποίηση μιας αξιόπιστης, ασφαλούς και λειτουργικά ολοκληρωμένης πλατφόρμας.

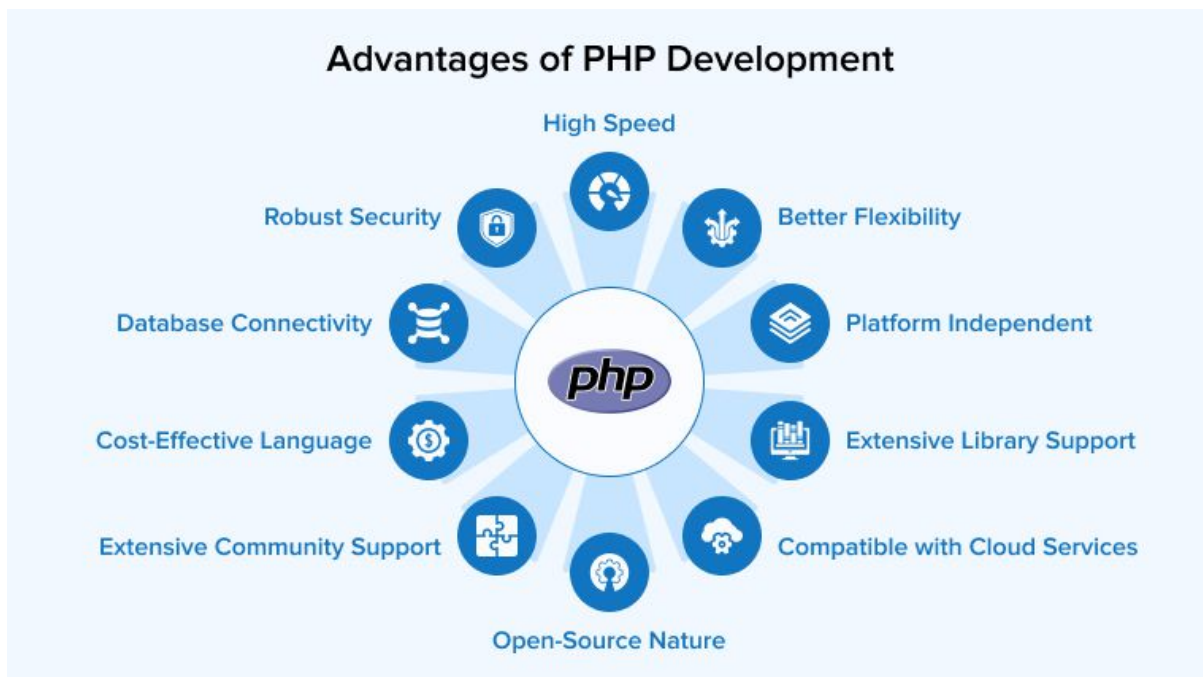
Η γλώσσα PHP ξεκίνησε το 1994 ως ένα προσωπικό project του Rasmus Lerdorf, με την αρχική της χρήση να περιορίζεται σε βασικές δυναμικές λειτουργίες για την παρακολούθηση επισκεψιμότητας προσωπικών ιστοσελίδων. Η πρώτη επίσημη έκδοση εμφανίστηκε το 1995 με την ονομασία PHP/FI (Personal Home Page / Forms Interpreter), παρέχοντας απλές δυνατότητες διαχείρισης φορμών και πρόσβασης σε βάσεις δεδομένων.

Με την πάροδο των ετών φαίνεται ότι η PHP εξελίχθηκε ραγδαία αποκτώντας όλο και περισσότερες δυνατότητες. Η έκδοση PHP 3.0 το 1998 αποτέλεσε σημείο καμπής εισάγοντας υποστήριξη για modular επέκταση μέσω API και δίνοντας στη γλώσσα πιο επαγγελματική κατεύθυνση. Η κυκλοφορία της PHP 4 το 2000 πρόσθεσε τον Zend Engine για δείξει σημαντικά την απόδοση και την υποστήριξη αντικειμενοστραφών τεχνικών.

Η σημαντικότερη αναβάθμιση έγινε με την PHP 5 το 2004 όταν η γλώσσα απέκτησε πλήρη υποστήριξη για αντικειμενοστραφή προγραμματισμό, μαζί με εργαλεία για καλύτερη διαχείριση εξαιρέσεων, υποστήριξη XML και βελτιωμένο χειρισμό βάσεων δεδομένων. Ακολούθησε η PHP 7 η οποία επικεντρώθηκε στη βελτίωση της απόδοσης για να μειωθεί το κόστος σε πόρους και αυξάνοντας την ταχύτητα εκτέλεσης κατά πολύ σε σχέση με τις προηγούμενες εκδόσεις.

Η πιο πρόσφατη σειρά εκδόσεων, PHP 8.x, εισήγαγε νέα χαρακτηριστικά όπως οι union types, attributes, JIT compiler και σημαντικές βελτιώσεις στη διαχείριση λαθών και την ασφάλεια. Η συνεχής εξέλιξη της γλώσσας κάνει την PHP σύγχρονη και ικανή να καλύψει τις ανάγκες απαιτητικών εφαρμογών.

Η χρήση της PHP στην ανάπτυξη διαδικτυακών εφαρμογών όπως η παρούσα πλατφόρμα διαχείρισης φοιτητικών εργασιών, έχει σημαντικά πλεονεκτήματα αλλά και ορισμένους περιορισμούς.[8]



Εικόνα 3.1: Key Advantages of PHP Development

[<https://www.tatvasoft.com/outsourcing/wp-content/uploads/2024/11/Advantages-of-PHP-Development.jpg>]

Ένα από τα σημαντικότερα πλεονεκτήματα είναι η ευκολία εκμάθησης και χρήσης της γλώσσας. Η PHP διαθέτει απλή και κατανοητή σύνταξη, γεγονός που την καθιστά ιδανική τόσο για αρχάριους προγραμματιστές όσο και για έμπειρους επαγγελματίες. Διαθέτει μια τεράστια και ενεργή κοινότητα, γεγονός που διασφαλίζει την ύπαρξη πλήθους βιβλιοθηκών, εργαλείων και πόρων υποστήριξης.

Ένα ακόμα πλεονέκτημα της PHP είναι η διαθεσιμότητα σε όλους τους δημοφιλείς web servers και λειτουργικά συστήματα. Πολλοί πάροχοι φιλοξενίας ιστοσελίδων υποστηρίζει PHP διευκολύνοντας την ανάπτυξη και εγκατάσταση διαδικτυακών εφαρμογών σε διάφορα περιβάλλοντα.

Ωστόσο, η PHP έχει μειονεκτήματα. Ένα από τα κυριότερα είναι η αρνητική φήμη που απέκτησε στο παρελθόν λόγω του μη οργανωμένου κώδικα και των ελλিপών πρακτικών ασφαλείας που συχνά ακολουθούσαν άπειροι προγραμματιστές. Παρότι οι σύγχρονες εκδόσεις της γλώσσας έχουν καλύψει σημαντικά αυτά τα κενά εξακολουθεί να απαιτείται ιδιαίτερη προσοχή από τον προγραμματιστή για την αποφυγή ευπαθειών και την εξασφάλιση καλής αρχιτεκτονικής του κώδικα.

Επιπλέον, σε εφαρμογές που απαιτούν πολύ υψηλές επιδόσεις ή επεξεργασία μεγάλου όγκου δεδομένων σε πραγματικό χρόνο η PHP δεν θεωρείται η βέλτιστη επιλογή αφού οι γλώσσες χαμηλότερου επιπέδου ή εξειδικευμένα περιβάλλοντα προσφέρουν καλύτερα αποτελέσματα.

Η PHP παραμένει μία από τις πιο ισχυρές επιλογές για την ανάπτυξη δυναμικών διαδικτυακών εφαρμογών ιδιαίτερα όταν συνδυάζεται με σύγχρονα frameworks, όπως το Laravel τα οποία επιβάλλουν βέλτιστες πρακτικές ανάπτυξης και βελτιώνουν τη δομή και την ασφάλεια του παραγόμενου λογισμικού.

3.2 Το Framework Laravel

Το Laravel αποτελεί ένα από τα πιο δημοφιλή και σύγχρονα frameworks για την ανάπτυξη διαδικτυακών εφαρμογών βασισμένων στη γλώσσα προγραμματισμού PHP. Δημιουργήθηκε το 2011 από τον Taylor Otwell με σκοπό να προσφέρει μια πιο κομψή απλή και ισχυρή λύση στην ανάπτυξη εφαρμογών και ξεπέρασε τους περιορισμούς που είχαν προηγούμενα frameworks και παραδοσιακές προσεγγίσεις στην PHP. Η φιλοσοφία του Laravel στηρίζεται στην απλότητα, την αναγνωσιμότητα του κώδικα και τη χρήση σύγχρονων πρακτικών ανάπτυξης και έγινε το ιδανικό για μικρές αλλά και ιδιαίτερα σύνθετες εφαρμογές. [9-12]

Το Laravel έκανε την εμφάνισή του το 2011 σε μια περίοδο όπου οι περισσότεροι προγραμματιστές PHP προσπαθούσαν να βρουν μια πιο κομψή και αποδοτική λύση για την ανάπτυξη εφαρμογών. Αν και υπήρχαν ήδη frameworks όπως το CodeIgniter, το Zend Framework και το Symfony, κανένα από αυτά δεν προσέφερε την απλότητα στη χρήση συνδυασμένη με ισχυρές σύγχρονες δυνατότητες. Ο Taylor Otwell, δημιουργός του Laravel, οραματίστηκε ένα framework που θα βασιζόταν σε ένα καθαρό και ευέλικτο API και ακολουθεί τα καλύτερα πρότυπα της σύγχρονης ανάπτυξης λογισμικού όπως το MVC (Model-View-Controller) και τα design patterns.

Η πρώτη έκδοση του Laravel (Laravel 1) περιλάμβανε βασικές δυνατότητες routing, sessions και authentication. Αλλά από την έκδοση Laravel 3 και μετά, το framework άρχισε να καθιερώνεται ως η κορυφαία επιλογή για ανάπτυξη σε PHP, εισάγοντας το Artisan Command-Line Interface, το Blade Template Engine και το πρώτο ενσωματωμένο σύστημα migration για τη διαχείριση βάσεων δεδομένων.

Το Laravel 4 αποτέλεσε σημαντική αναβάθμιση, βασισμένη πλήρως στο Composer, το διαχειριστή πακέτων της PHP και διευκολύνει την εγκατάσταση και χρήση έτοιμων βιβλιοθηκών. Η έλευση του Laravel 5 το 2015 προσέφερε ακόμα περισσότερες βελτιώσεις στην οργάνωση του κώδικα και ενίσχυσε το σύστημα routing, τα middleware με ισχυρούς μηχανισμούς dependency injection.

Με τις τελευταίες εκδόσεις Laravel 9 και 10 το framework έχει εξελιχθεί σε ένα από τα πιο πλήρη και αξιόπιστα εργαλεία ανάπτυξης web εφαρμογών και υποστηρίζει σύγχρονες τεχνικές όπως event broadcasting, job queues, API resources και προηγμένες τεχνολογίες ασφάλειας. Και συνεχίζει να δίνει έμφαση στη φιλικότητα προς τον προγραμματιστή και τη συνεχή βελτίωση της απόδοσης.

Η αρχιτεκτονική του Laravel βασίζεται στο πρότυπο MVC (Model-View-Controller) το οποίο προσφέρει έναν σαφή διαχωρισμό των επιπέδων της εφαρμογής. Παρόλο που στην παρούσα εργασία επιλέχθηκε να μην χρησιμοποιηθεί το Eloquent ORM για την επικοινωνία με τη βάση δεδομένων, αξιοποιήθηκαν πλήρως τα υπόλοιπα στοιχεία του framework όπως η ισχυρή διαχείριση routes, το σύστημα templating Blade και η ευκολία χειρισμού αιτημάτων HTTP μέσω Controllers. Το Laravel διευκολύνει την ανάπτυξη οργανωμένων και επεκτάσιμων εφαρμογών για καλύτερη δομή στον κώδικα και βελτιώνει την αναγνωσιμότητά του.

Το Laravel χρησιμοποιήθηκε ως η κύρια πλατφόρμα ανάπτυξης του backend της εφαρμογής. Μέσω του μηχανισμού routing του Laravel, υλοποιήθηκαν όλες οι βασικές διαδρομές (routes) της εφαρμογής, τόσο για τους φοιτητές όσο και για τους καθηγητές και διαχειριστές. Αυτό επέτρεψε την απλοποιημένη διαχείριση των αιτημάτων προς τον server, την άμεση επεξεργασία τους από τα αντίστοιχα Controllers και την ομαλή μετάβαση ανάμεσα στις διαφορετικές λειτουργίες του συστήματος.

Το Blade μας το ενσωματωμένο template engine του Laravel διευκόλυνε την κατασκευή της διεπαφής χρήστη. Μέσω της χρήσης Blade templates πέτυχε η δημιουργία ενός καθαρού, εύχρηστου και επαναχρησιμοποιήσιμου UI και μειώνει σημαντικά τον επιπλέον κώδικα και διευκολύνει την ενσωμάτωση δυναμικού περιεχομένου μέσα στις ιστοσελίδες. Η δυνατότητα να χρησιμοποιούνται έτοιμες δομές layout και components σε συνδυασμό με τις Blade directives έχει συνεβάλλει στην επιτάχυνση της ανάπτυξης και στη βελτίωση της συντηρησιμότητας του έργου.

Σημαντικό ρόλο στην υλοποίηση έπαιξε επίσης το σύστημα διαχείρισης sessions και authentication που προσφέρει το Laravel. Η πλατφόρμα υποστηρίζει ενσωματωμένους μηχανισμούς για την ασφαλή αποθήκευση sessions και την προστασία από κοινές επιθέσεις όπως CSRF και XSS. Στο παρόν έργο μέσω του Laravel, διαχειρίστηκαν με ευκολία τα sessions για την αναγνώριση και διατήρηση των στοιχείων των χρηστών καθ' όλη τη διάρκεια της πλοήγησης και εξασφαλίζουμε ότι κάθε ενέργεια γίνεται από αυθεντικοποιημένους χρήστες με τα κατάλληλα δικαιώματα.

Αν και δεν χρησιμοποιήθηκε το Eloquent ORM για την αλληλεπίδραση με τη βάση δεδομένων το Laravel παρέχει το ισχυρό DB Facade και χρησιμοποιήθηκε εκτενώς για την εκτέλεση απευθείας SQL ερωτημάτων με ασφαλή και αποδοτικό τρόπο. Μέσω αυτής της προσέγγισης η εφαρμογή είχε υψηλή απόδοση και πλήρη έλεγχο στη δομή των ερωτημάτων προς τη βάση δεδομένων MySQL χωρίς την ανάγκη επιπλέον abstraction layers.

Το Laravel είναι το θεμέλιο πάνω στο οποίο στηρίχθηκε η ανάπτυξη της διαδικτυακής πλατφόρμας. Συνδυάζοντας την καθαρότητα του κώδικα με την ισχυρή υποστήριξη της κοινότητας, την εκτεταμένη

τεκμηρίωση και τα προηγμένα χαρακτηριστικά ασφαλείας το framework αυτό συνέβαλε στην υλοποίηση ενός συστήματος φιλικού προς τον χρήστη και εύκολα επεκτάσιμου για μελλοντικές βελτιώσεις και προσθήκες.

Το Laravel έχει καθιερωθεί ως ένα από τα κορυφαία frameworks κυρίως λόγω της ισορροπίας που προσφέρει μεταξύ απλότητας και ισχύος. Ένα από τα βασικότερα πλεονεκτήματά του είναι η σαφής δομή και η καθαρότητα του κώδικα που επιβάλλει μέσω του προτύπου MVC. Οι προγραμματιστές μπορούν να διαχωρίζουν εύκολα τα δεδομένα, τη λογική της εφαρμογής και την παρουσίαση, οδηγώντας σε πιο συντηρήσιμο και επεκτάσιμο κώδικα.

Η ύπαρξη του Artisan CLI προσφέρει σημαντική επιτάχυνση κατά την ανάπτυξη, διευκολύνοντας την αυτοματοποίηση επαναλαμβανόμενων εργασιών όπως η δημιουργία controllers, models και migrations. Το Blade Template Engine επιτρέπει την ανάπτυξη ευέλικτων και καθαρών views με ελάχιστο κόπο και διευκολύνει την ενσωμάτωση δυναμικού περιεχομένου χωρίς την πολυπλοκότητα άλλων template engines.

Στον τομέα της ασφάλειας το Laravel παρέχει ενσωματωμένους μηχανισμούς για την προστασία από κοινές ευπάθειες, όπως CSRF, XSS και SQL Injection. Μέσω του middleware συστήματος πετυγχένεται εύκολα ο έλεγχος πρόσβασης και η διαχείριση των ρόλων των χρηστών.

Το Laravel παρουσιάζει και ορισμένα μειονεκτήματα. Η πολυπλοκότητα του framework αυξάνεται σημαντικά σε πολύ μεγάλες εφαρμογές και θέλει υψηλότερο επίπεδο εμπειρίας από τον προγραμματιστή για την αποτελεσματική του χρήση. Η αποφυγή του Eloquent ORM όπως στην παρούσα εργασία βοηθά στην καλύτερη διαχείριση της απόδοσης και μειώνει ορισμένα από τα πλεονεκτήματα ταχείας ανάπτυξης που προσφέρει το framework.

Λόγω του πλήθους των διαθέσιμων χαρακτηριστικών υπάρχει το ρίσκο υπερφόρτωσης του έργου με περιττές λειτουργίες αν δεν υπάρχει σωστός σχεδιασμός από την αρχή. Το Laravel παραμένει μία από τις πιο δημοφιλείς επιλογές για την ανάπτυξη διαδικτυακών εφαρμογών και έχει ένα ισχυρό και ταυτόχρονα φιλικό προς τον προγραμματιστή περιβάλλον ανάπτυξης.

3.3 MySQL

Η MySQL είναι ένα από τα πιο διαδεδομένα και αξιόπιστα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) παγκοσμίως. Δημιουργήθηκε το 1995 από την εταιρεία MySQL AB και αργότερα εξαγοράστηκε από την Oracle Corporation. Από την αρχή της κυκλοφορίας της μέχρι σήμερα, η MySQL έχει καθιερωθεί ως μια σταθερή, αποδοτική και εύκολη στη χρήση λύση για την

αποθήκευση, διαχείριση και ανάκτηση δεδομένων για σε μικρές εφαρμογές όσο και σε πολύπλοκα και απαιτητικά πληροφοριακά συστήματα. [13-15]

Η ευρεία διάδοσή της οφείλεται στην υψηλή απόδοσή της στη δωρεάν διάθεσή της μέσω της κοινότητας (Community Edition), αλλά και στη συμβατότητά της με τις πιο γνωστές γλώσσες προγραμματισμού, όπως PHP, Python, Java, C++ και πολλές άλλες. Η MySQL υιοθετεί το πρότυπο SQL (Structured Query Language) για την αλληλεπίδραση με τις βάσεις δεδομένων και διευκολύνει την εκτέλεση πολύπλοκων ερωτημάτων, την οργάνωση των δεδομένων μέσω σχέσεων και τη δημιουργία ολοκληρωμένων επιχειρησιακών μοντέλων.

Στην παρούσα εργασία η MySQL επιλέχθηκε για την αποθήκευση όλων των κρίσιμων δεδομένων της πλατφόρμας διαχείρισης φοιτητικών εργασιών. Η απόφαση αυτή βασίστηκε στη σταθερότητα και την αξιοπιστία που προσφέρει το σύστημα και στη στενή ενσωμάτωσή του με το οικοσύστημα της PHP και του Laravel. Η βάση δεδομένων φιλοξενεί όλα τα απαραίτητα δεδομένα που σχετίζονται με τους χρήστες, τα μαθήματα, τις ανατιθέμενες εργασίες, τις υποβολές φοιτητών και τις βαθμολογίες.

Η MySQL επιτρέπει την εύκολη δημιουργία και διαχείριση πινάκων χωρίς την ανάγκη πολύπλοκων εργαλείων ή διαδικασιών. Στην υλοποίηση της συγκεκριμένης πλατφόρμας ακολουθήθηκε μια απλή αλλά λειτουργική σχεδίαση της βάσης και αποφεύγει τη χρήση Foreign Keys, προκειμένου να διατηρηθεί η ευελιξία στη διαχείριση των δεδομένων και να διευκολυνθεί η άμεση εκτέλεση ερωτημάτων μέσω του DB Facade στο Laravel.

Μέσω της MySQL επιτυγχάνεται γρήγορη ανάκτηση και επεξεργασία των δεδομένων και η γλώσσα SQL παρέχει τη δυνατότητα εκτέλεσης πολύπλοκων ερωτημάτων που καλύπτουν τις ανάγκες προβολής, φιλτραρίσματος και αναλυτικής επεξεργασίας των πληροφοριών. Για παράδειγμα οι αναζητήσεις για τα μαθήματα στα οποία συμμετέχει ένας φοιτητής, οι εργασίες που έχει υποβάλει, καθώς και οι υπολογισμοί μέσω όρων βαθμολογιών πραγματοποιούνται άμεσα με τη χρήση αποδοτικών SQL εντολών.

Η MySQL διαθέτει επίσης ενσωματωμένους μηχανισμούς για την προστασία των δεδομένων, όπως έλεγχο πρόσβασης μέσω χρηστών και δικαιωμάτων, backup και recovery δυνατότητες και υποστήριξη για transactions όταν αυτό απαιτείται για τη διασφάλιση της ακεραιότητας των δεδομένων.

Η χρήση της MySQL στο συγκεκριμένο έργο μας έδωσε την αποθήκευση και διαχείριση μεγάλου όγκου δεδομένων με απόλυτη αξιοπιστία και ταχύτητα. Η δυνατότητα άμεσης σύνδεσης με την PHP μέσω του Laravel DB Facade επέτρεψε την εκτέλεση ερωτημάτων με ασφαλή και ελεγχόμενο τρόπο, χωρίς την ανάγκη χρήσης πολύπλοκων layers.

Η MySQL αποτέλεσε μια από τις βασικές τεχνολογίες για την επιτυχία της πλατφόρμας. Με υψηλή απόδοση, ευκολία στη χρήση και διαχείριση, αλλά και δυνατότητες που καλύπτουν πλήρως τις

απαιτήσεις ενός σύγχρονου πληροφοριακού συστήματος, η MySQL παραμένει μία από τις πλέον αξιόπιστες λύσεις για την αποθήκευση και διαχείριση δεδομένων σε διαδικτυακές εφαρμογές.

Η MySQL, ως ένα από τα πιο διαδεδομένα συστήματα διαχείρισης βάσεων δεδομένων προσφέρει μια σειρά από σημαντικά πλεονεκτήματα, αλλά και ορισμένα μειονεκτήματα που πρέπει να λαμβάνονται υπόψη κατά το σχεδιασμό και την ανάπτυξη ενός συστήματος.

Η ευρεία διάδοση της MySQL και η μεγάλη υποστηρικτική κοινότητα αποτελούν σημαντικά πλεονεκτήματα, καθώς διασφαλίζουν διαρκή ανάπτυξη, επίλυση προβλημάτων και εκτενή τεκμηρίωση. Το γεγονός ότι είναι ένα ανοιχτού κώδικα σύστημα (στην Community Edition) επιτρέπει τη χρήση της χωρίς κόστος για μικρές και μεσαίες εφαρμογές και προσφέρει και εμπορικές εκδόσεις με πρόσθετες δυνατότητες για πιο απαιτητικά περιβάλλοντα.

Η MySQL διακρίνεται για την υψηλή της απόδοση ειδικά σε εφαρμογές ανάγνωσης δεδομένων και σε περιβάλλοντα όπου δεν απαιτείται ιδιαίτερα πολύπλοκη λογική συναλλαγών. Η συμβατότητά της με όλες τις μεγάλες γλώσσες προγραμματισμού και πλατφόρμες την καθιστά ευέλικτη και εύκολα ενσωματώσιμη σε υπάρχουσες υποδομές.

Σημαντικό πλεονέκτημα αποτελεί και η απλότητα στη διαχείριση. Η εγκατάσταση, η διαμόρφωση και η καθημερινή διαχείριση της βάσης δεδομένων πραγματοποιούνται εύκολα ενώ υπάρχουν διαθέσιμα γραφικά εργαλεία, όπως το MySQL Workbench, για την οπτική απεικόνιση και τη διαχείριση της δομής της βάσης.

Παρά τα πολλά πλεονεκτήματά της, η MySQL παρουσιάζει ορισμένους περιορισμούς κυρίως σε εφαρμογές που απαιτούν αυστηρή διαχείριση συναλλαγών και πολύπλοκους περιορισμούς. Αν και υποστηρίζει transactions, η χρήση τους δεν είναι τόσο εκτενής και ισχυρή όσο σε πιο προηγμένα RDBMS, όπως το PostgreSQL ή το Oracle DB.

Η έλλειψη υποστήριξης για ορισμένα advanced features (όπως πλήρη υποστήριξη για stored procedures σε παλαιότερες εκδόσεις ή περιορισμένες δυνατότητες σε triggers και events) περιορίζει την αξιοποίησή της σε εφαρμογές με αυξημένες ανάγκες πολύπλοκης επιχειρησιακής λογικής σε επίπεδο βάσης.

Ένα ακόμα μειονέκτημα ειδικά για τα μεγάλα συστήματα είναι η περιορισμένη υποστήριξη πλήρους consistency σε περιβάλλοντα υψηλής ταυτόχρονης πρόσβασης. Εκτός και αν γίνει πολύ προσεκτικός σχεδιασμός του συστήματος και σωστή χρήση των κατάλληλων storage engines.

Στην υλοποίηση του έργου δεν χρησιμοποιήθηκε το Eloquent ORM αλλά το Laravel προσφέρει τη δυνατότητα εκτέλεσης άμεσων και αποδοτικών SQL ερωτημάτων μέσω του DB Facade, παρέχοντας πλήρη έλεγχο στον προγραμματιστή.

1. Ανάκτηση δεδομένων (SELECT):

```
use Illuminate\Support\Facades\DB;

$students = DB::table('users')
    ->where('kind', 0) // 0 = Φοιτητής
    ->get();
```

2. Εισαγωγή νέας εγγραφής (INSERT):

```
DB::table('courses')->insert([
    'name' => 'Προγραμματισμός Γ',
    'description' => 'Εισαγωγή στον προγραμματισμό.',
    'created_at' => now()
]);
```

3. Ενημέρωση εγγραφής (UPDATE):

```
DB::table('courseworks')
    ->where('id', 5)
    ->update(['deadlineend' => '2024-06-30']);
```

4. Διαγραφή εγγραφής (DELETE):

```
DB::table('works')
    ->where('id', 12)
    ->delete();
```

5. Εκτέλεση ακατέργαστου SQL ερωτήματος (Raw Query):

```
$results = DB::select('SELECT * FROM users WHERE email = ?', ['user@example.com']);
```

Αυτά τα παραδείγματα δείχνουν πόσο απλή και ισχυρή είναι η αλληλεπίδραση του Laravel με τη MySQL ακόμα και χωρίς τη χρήση ORM. Η προσέγγιση αυτή διευκολύνει την ανάπτυξη εφαρμογών με αυξημένο έλεγχο στην εκτέλεση των ερωτημάτων και στη βελτιστοποίηση της απόδοσης ιδανική για εφαρμογές όπως η παρούσα πλατφόρμα διαχείρισης φοιτητικών εργασιών.

Κεφάλαιο 4ο: Η Διαδικτυακή Πλατφόρμα Διαχείρισης Εργασιών

4.1 Εισαγωγή – Η διαδικασία

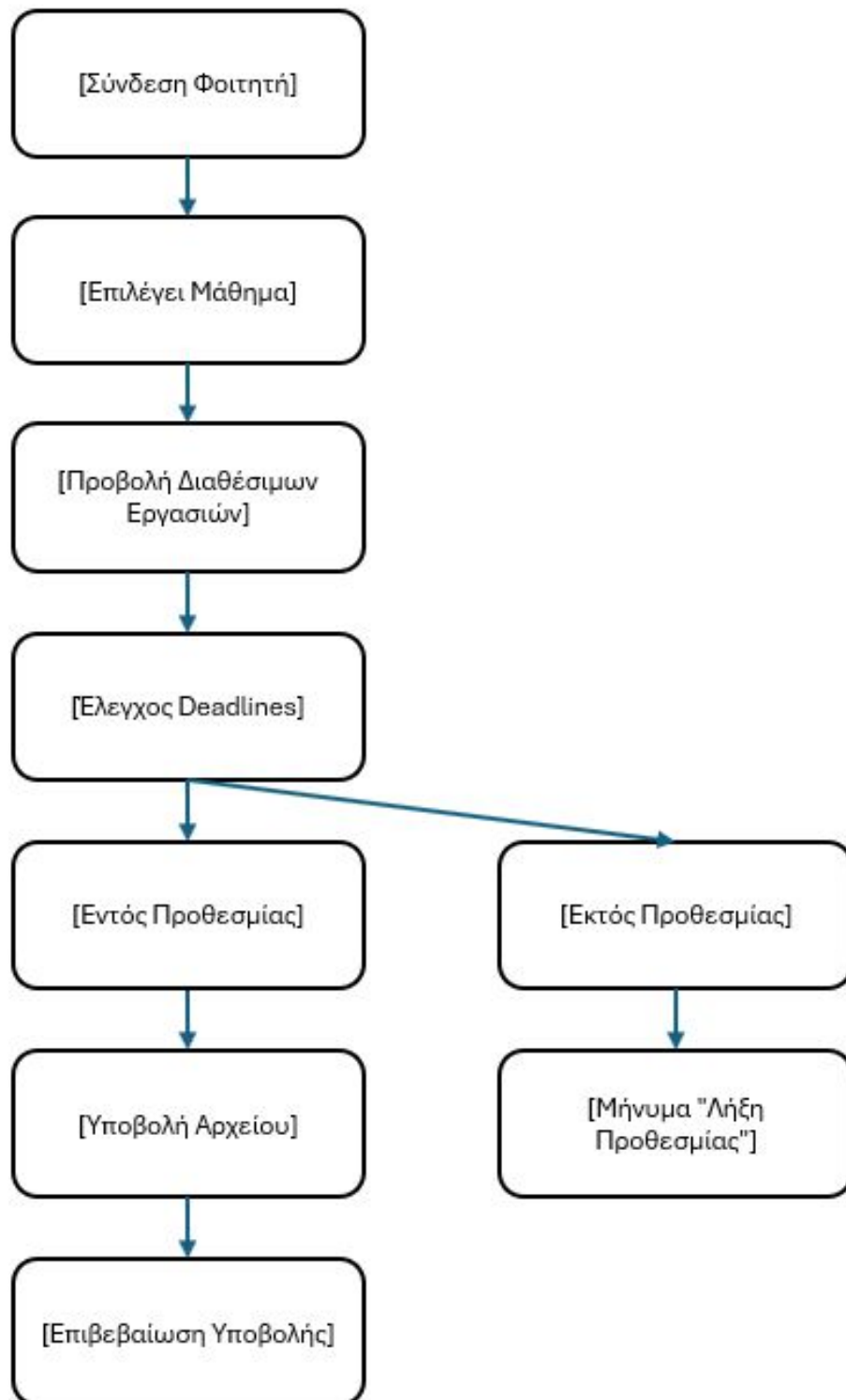
Η ανάπτυξη διαδικτυακών πλατφορμών για την υποστήριξη ακαδημαϊκών διαδικασιών αποτελεί ένα κομμάτι της σύγχρονης εκπαιδευτικής πρακτικής. Η παρούσα εργασία παρουσιάζει την υλοποίηση μιας ολοκληρωμένης διαδικτυακής πλατφόρμας διαχείρισης φοιτητικών εργασιών η οποία επιτρέπει την ηλεκτρονική υποβολή, αξιολόγηση και παρακολούθηση της προόδου των φοιτητών.

Η πλατφόρμα έχει σχεδιαστεί με στόχο την απλοποίηση των διαδικασιών τόσο για τους φοιτητές όσο και για τους καθηγητές, παρέχοντας ένα φιλικό και εύχρηστο περιβάλλον. Οι φοιτητές μπορούν να υποβάλλουν τις εργασίες τους ηλεκτρονικά, να παρακολουθούν την πορεία αξιολόγησής τους και να λαμβάνουν σχόλια από τους διδάσκοντες. Παράλληλα, οι καθηγητές έχουν τη δυνατότητα να διαχειρίζονται τα μαθήματα και τις εργασίες τους, να αξιολογούν τις υποβολές και να παρέχουν εξατομικευμένα σχόλια και βαθμολογίες.

Η υλοποίηση της πλατφόρμας πραγματοποιήθηκε χρησιμοποιώντας το framework Laravel σε συνδυασμό με MySQL για την αποθήκευση των δεδομένων και το Bootstrap για την ανάπτυξη ενός responsive και μοντέρνου γραφικού περιβάλλοντος. Ιδιαίτερη έμφαση δόθηκε στη σωστή διαχείριση των ρόλων χρηστών (φοιτητές, καθηγητές, διαχειριστές) και στην εξασφάλιση της ακεραιότητας των δεδομένων μέσω ελέγχων κατά την υποβολή και αξιολόγηση των εργασιών.

Η πλατφόρμα αυτή μπορεί να χρησιμοποιηθεί ως βάση για την επέκταση και υλοποίηση πιο σύνθετων συστημάτων διαχείρισης εκπαιδευτικού περιεχομένου.

4.2 Η διαδικασία



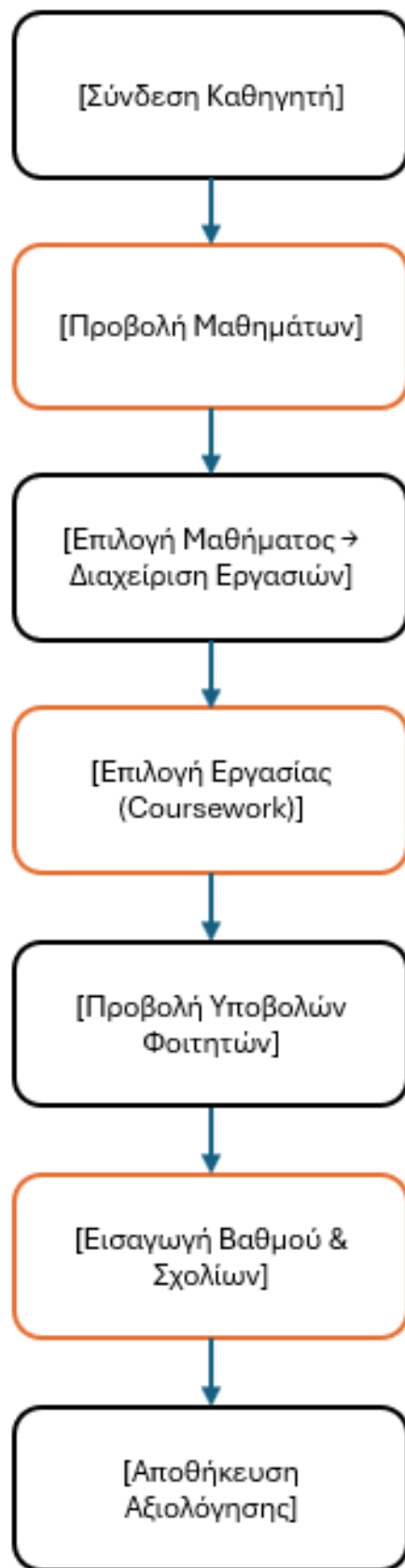
Εικόνα 4.1: Διάγραμμα Διαδικασίας Υποβολής Εργασίας (Student Workflow)

Η διαδικασία υποβολής εργασίας ξεκινά με την είσοδο του φοιτητή στην πλατφόρμα και την επιλογή του μαθήματος στο οποίο είναι εγγεγραμμένος. Στη συνέχεια εμφανίζεται η λίστα με τις διαθέσιμες εργασίες (courseworks) του μαθήματος. Το σύστημα πραγματοποιεί αυτόματο έλεγχο των χρονικών ορίων (deadlines) της κάθε εργασίας.

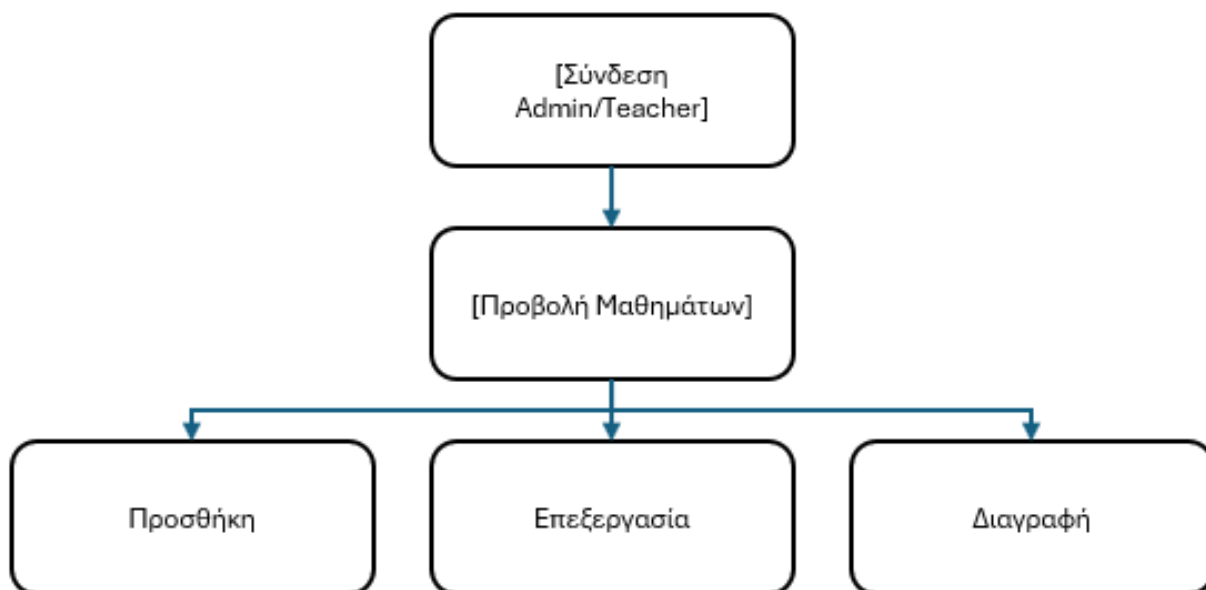
Εάν η προθεσμία δεν έχει ξεκινήσει ή έχει λήξει, ο φοιτητής λαμβάνει σχετικό ενημερωτικό μήνυμα και δεν του παρέχεται η δυνατότητα υποβολής. Αν η εργασία βρίσκεται εντός του επιτρεπόμενου χρονικού διαστήματος, ο φοιτητής μπορεί να ανεβάσει το αρχείο της εργασίας του και να λάβει επιβεβαίωση επιτυχούς υποβολής.

Ο καθηγητής μέσω του προσωπικού του λογαριασμού, έχει τη δυνατότητα να διαχειριστεί τα μαθήματα που του έχουν ανατεθεί. Για κάθε μάθημα μπορεί να προβάλλει τη λίστα των διαθέσιμων εργασιών και τις υποβολές των φοιτητών.

Η διαδικασία αξιολόγησης περιλαμβάνει την αναλυτική επισκόπηση των υποβληθέντων αρχείων και την καταχώρηση βαθμολογίας και σχολίων για κάθε υποβολή. Η βαθμολογία και τα σχόλια αποθηκεύονται στο σύστημα, παρέχοντας άμεση ανατροφοδότηση στους φοιτητές.

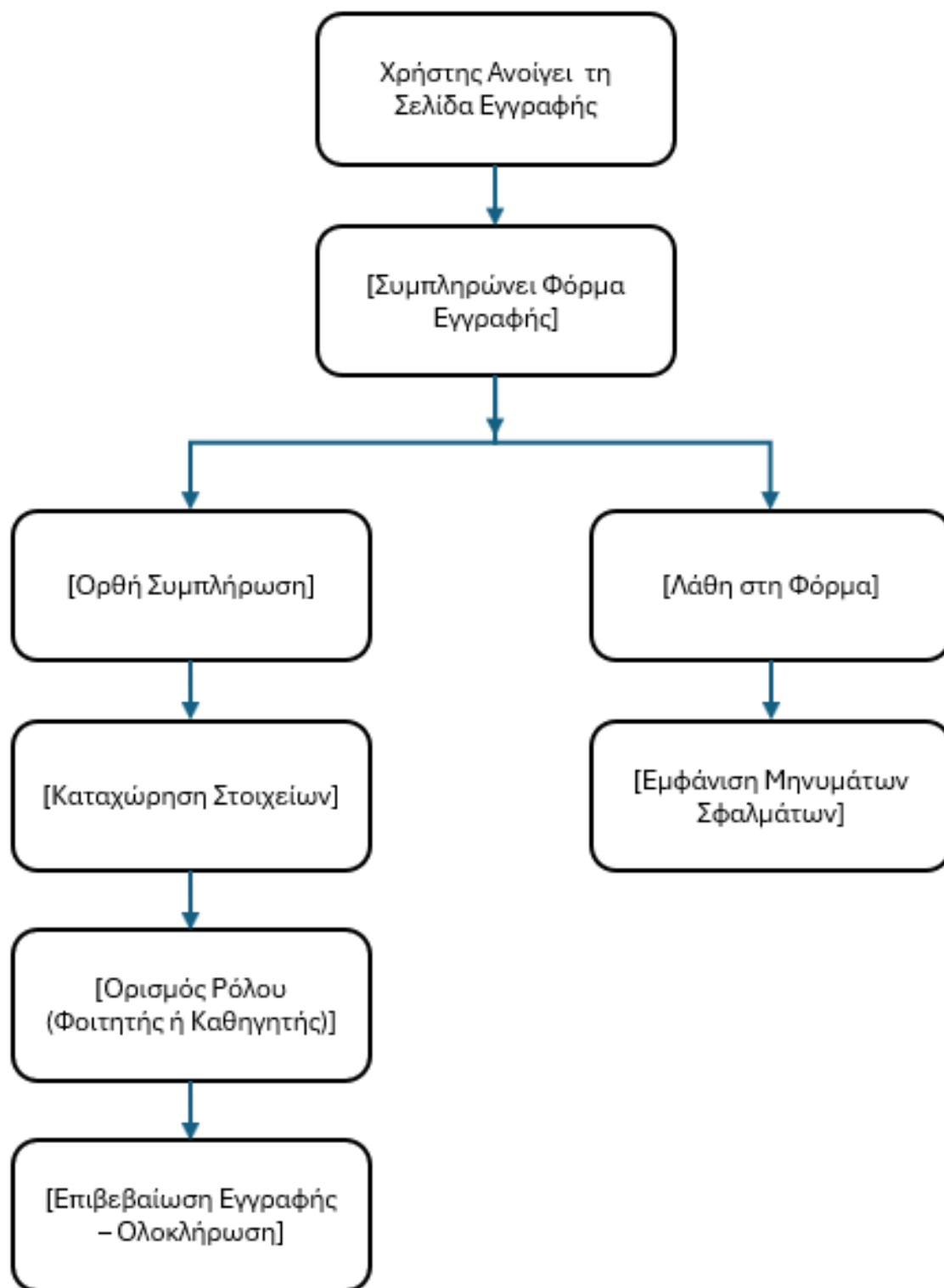


Εικόνα 4.2: Διάγραμμα Διαδικασίας Βαθμολόγησης Εργασιών (Teacher Workflow)



Εικόνα 4.3: Διάγραμμα Διαχείρισης Μαθημάτων (Admin/Teacher)

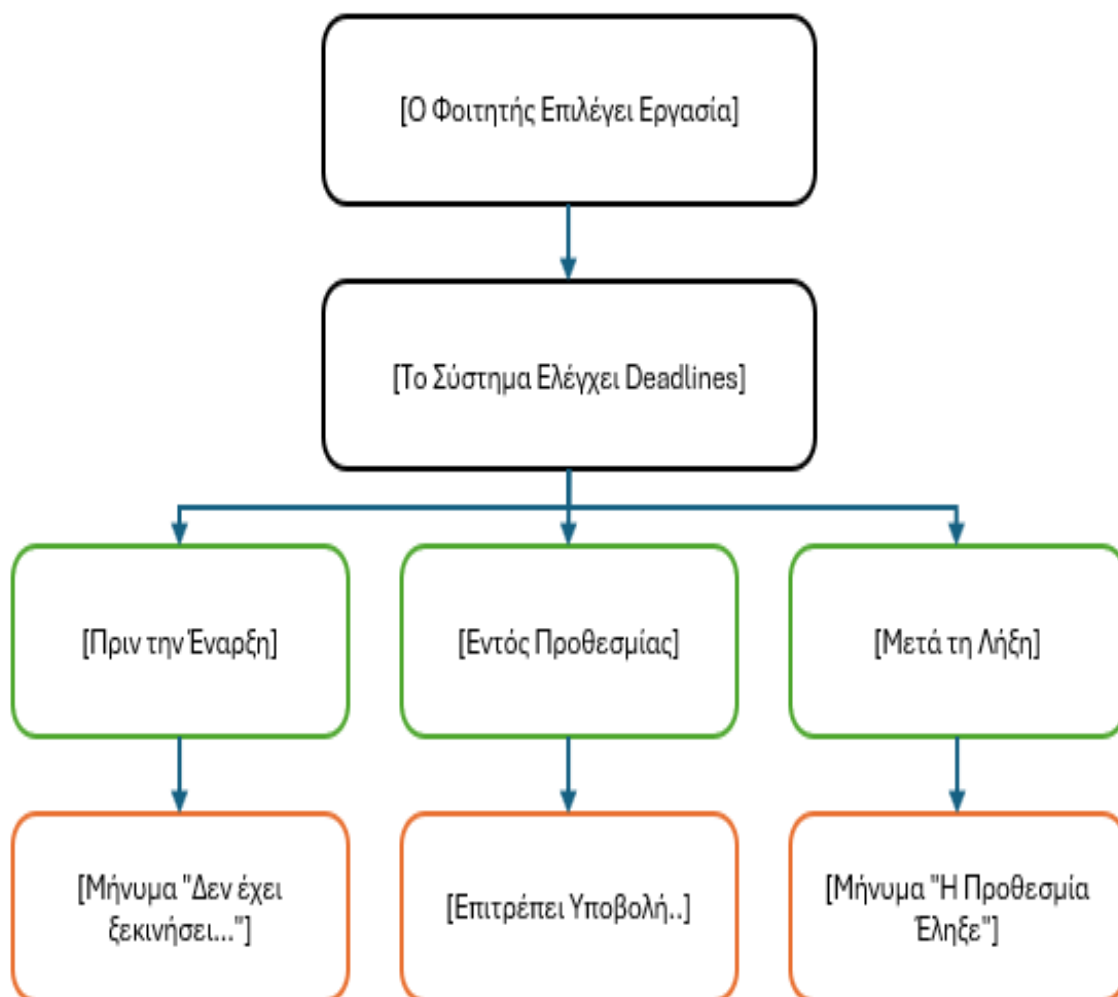
Οι διαχειριστές και οι καθηγητές έχουν πρόσβαση σε λειτουργίες διαχείρισης μαθημάτων. Μέσω της πλατφόρμας μπορούν να δημιουργούν νέα μαθήματα, να επεξεργάζονται τις πληροφορίες τους (τίτλος, περιγραφή, υπεύθυνος καθηγητής) και να τα διαγράφουν, εφόσον χρειάζεται. Η διαδικασία αυτή απλοποιεί τη διαχείριση της εκπαιδευτικής δραστηριότητας και επιτρέπει την ορθή αντιστοίχιση μαθημάτων με τους αντίστοιχους διδάσκοντες.



Εικόνα 4.4: Διάγραμμα Διαδικασίας Εγγραφής Νέου Φοιτητή/Καθηγητή

Η πλατφόρμα παρέχει τη δυνατότητα εγγραφής νέων χρηστών, είτε φοιτητών είτε καθηγητών. Ο χρήστης μεταβαίνει στη φόρμα εγγραφής όπου εισάγει τα απαραίτητα στοιχεία (ονοματεπώνυμο, email, κωδικό πρόσβασης κ.λπ.).

Το σύστημα πραγματοποιεί έλεγχο εγκυρότητας των δεδομένων και, αν εντοπιστούν λάθη, εμφανίζονται κατάλληλα μηνύματα για διόρθωση. Μετά την επιτυχή καταχώρηση των στοιχείων, ο χρήστης αποκτά πρόσβαση στο σύστημα με τα αντίστοιχα δικαιώματα ανάλογα με το ρόλο του.



Εικόνα 4.5: Διάγραμμα Διαδικασίας Ελέγχου Deadlines και Πρόσβασης Υποβολής (Ασφάλεια Υποβολής)

Ο έλεγχος των χρονικών ορίων αποτελεί κρίσιμο μηχανισμό ασφαλείας στην πλατφόρμα. Πριν επιτραπεί η υποβολή εργασίας από τον φοιτητή, το σύστημα ελέγχει αν η τρέχουσα ημερομηνία

βρίσκεται εντός του επιτρεπόμενου διαστήματος υποβολής. Εάν η προθεσμία δεν έχει ξεκινήσει ακόμα, εμφανίζεται μήνυμα που ενημερώνει τον φοιτητή ότι η υποβολή δεν είναι ακόμη διαθέσιμη. Αν η προθεσμία έχει λήξει, εμφανίζεται μήνυμα ότι η υποβολή έχει κλείσει και η διαδικασία υποβολής απαγορεύεται. Αυτός ο έλεγχος διασφαλίζει τη σωστή τήρηση των κανόνων και την ακεραιότητα της διαδικασίας αξιολόγησης.

Πίνακας 1. συνοπτικός πίνακας πρόσβασης ανά ρόλο

Διαδικασία	Φοιτητής	Καθηγητής	Admin
Προβολή Μαθημάτων	ΝΑΙ	ΝΑΙ	ΝΑΙ
Εγγραφή σε Μάθημα	ΟΧΙ	ΝΑΙ	ΝΑΙ
Υποβολή Εργασίας (εντός deadline)	ΝΑΙ	ΟΧΙ	ΟΧΙ
Προβολή Βαθμών	ΝΑΙ	ΟΧΙ	ΟΧΙ
Διαχείριση Μαθημάτων	ΟΧΙ	ΝΑΙ	ΝΑΙ
Δημιουργία/Επεξεργασία Εργασιών	ΟΧΙ	ΝΑΙ	ΝΑΙ
Προβολή Υποβολών Φοιτητών	ΟΧΙ	ΝΑΙ	ΟΧΙ
Βαθμολόγηση Υποβολών	ΟΧΙ	ΝΑΙ	ΟΧΙ
Διαχείριση Χρηστών (Εγγραφή/Ρόλοι)	ΟΧΙ	ΟΧΙ	ΝΑΙ
Διαχείριση Προθεσμιών Εργασιών	ΟΧΙ	ΝΑΙ	ΝΑΙ

- Η εγγραφή φοιτητών σε μαθήματα γίνεται από καθηγητές ή admin, όχι απευθείας από φοιτητές.
- Ο Admin διαχειρίζεται κυρίως δομές του συστήματος (μαθήματα, χρήστες) και όχι αξιολογήσεις.
- Οι προθεσμίες εργασιών ρυθμίζονται από καθηγητές κατά τη δημιουργία/επεξεργασία εργασιών.

Η πλατφόρμα βασίζεται σε μια τυπική αρχιτεκτονική MVC (Model-View-Controller), η οποία διασφαλίζει την οργάνωση του κώδικα, τη διαχωρισμένη διαχείριση των δεδομένων και την ευκολία στη συντήρηση και επέκταση του συστήματος.

Model (Δεδομένα & Δομή Βάσης)

- Η διαχείριση των δεδομένων γίνεται μέσω απευθείας κλήσεων στη βάση με τη χρήση του DB Facade (χωρίς Eloquent Models).
- Οι κύριοι πίνακες της βάσης είναι:
 - users: Χρήστες της πλατφόρμας (φοιτητές, καθηγητές, admins).
 - courses: Μαθήματα.
 - course_student: Συσχέτιση φοιτητών με μαθήματα.
 - courseworks: Εργασίες ανά μάθημα.
 - works: Υποβολές εργασιών από φοιτητές.

Controller (Controller Λογική)

- Υλοποιούνται controllers για την εξυπηρέτηση διαφορετικών ρόλων:
 - UserController: Διαχείριση αυθεντικοποίησης και dashboard.
 - StudentController: Διαχείριση λειτουργιών φοιτητών (υποβολές, προβολή μαθημάτων, βαθμολογίες).
 - TeacherController: Διαχείριση μαθημάτων, φοιτητών, εργασιών και βαθμολογιών.
 - AdminController: Πλήρης διαχείριση μαθημάτων και χρηστών.

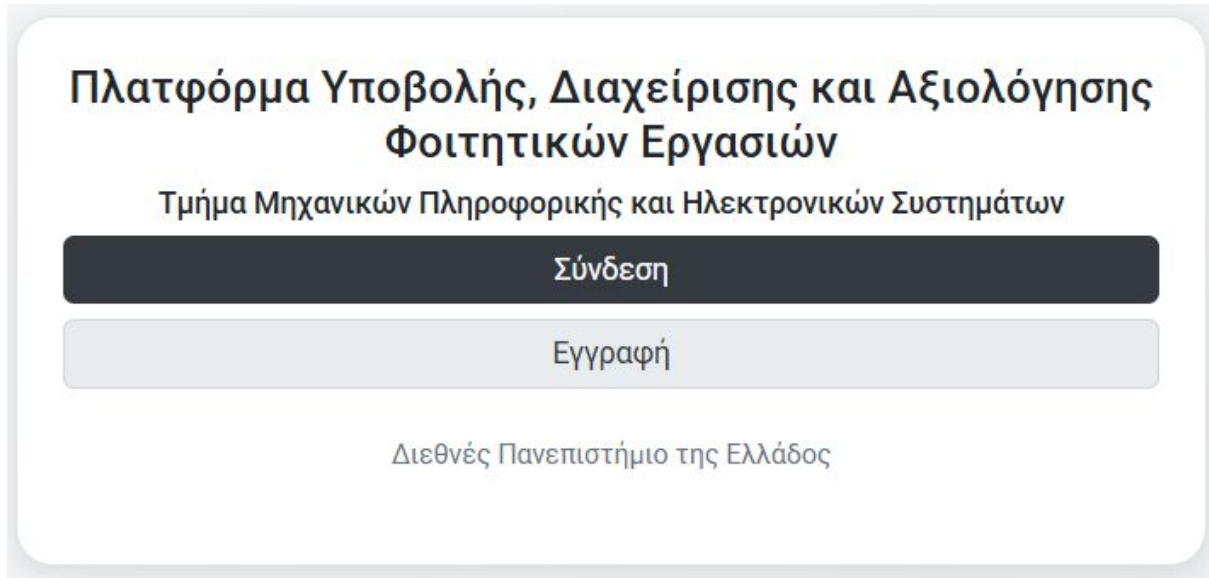
View (Χρήστης & Διεπαφή)

- Η διεπαφή υλοποιείται με **Blade Templates** και χρήση του **Bootstrap** για responsive σχεδίαση.
- Κάθε ρόλος έχει προσαρμοσμένο dashboard και σελίδες ανάλογα με τα διαθέσιμα δικαιώματά του.

Ασφάλεια & Έλεγχοι Πρόσβασης

- Η πρόσβαση σε λειτουργίες γίνεται βάσει του ρόλου που έχει αποθηκευτεί στο Session (kind στο users table).
- Υπάρχουν έλεγχοι στο Blade και στον Controller ώστε να αποτρέπεται η παράκαμψη των κανόνων, ειδικά στις υποβολές εργασιών.

4.3 Διαδικτυακή εφαρμογή - ιστοσελίδα



Εικόνα 4.6: Η πρώτη σελίδα

Η εικόνα 4.6 παρουσιάζει την αρχική σελίδα της πλατφόρμας διαχείρισης φοιτητικών εργασιών. Από αυτήν την οθόνη, ο χρήστης μπορεί να επιλέξει να συνδεθεί στον λογαριασμό του ή να προχωρήσει στη δημιουργία νέου λογαριασμού μέσω της επιλογής εγγραφής. Η σχεδίαση της σελίδας είναι απλή και λειτουργική, επιτρέποντας στον επισκέπτη να εντοπίσει άμεσα τις βασικές επιλογές πλοήγησης. Παράλληλα, δίνεται έμφαση στη φιλικότητα προς τον χρήστη και στην καθαρότητα της παρουσίασης των διαθέσιμων ενεργειών.

Πλατφόρμα Υποβολής, Διαχείρισης και
Αξιολόγησης Φοιτητικών Εργασιών
Σύνδεση

Email

Κωδικός

Συνδεθείτε

[Εγγραφή](#)

Εικόνα 4.7: Σύνδεση

Στην εικόνα 4.7 απεικονίζεται η σελίδα σύνδεσης της πλατφόρμας, μέσω της οποίας οι χρήστες εισάγουν τα διαπιστευτήριά τους για να αποκτήσουν πρόσβαση στις διαθέσιμες λειτουργίες. Η φόρμα σύνδεσης περιλαμβάνει πεδία για την εισαγωγή email και κωδικού πρόσβασης, καθώς και επιλογές για την υποβολή της φόρμας και την ανάκτηση κωδικού σε περίπτωση που ο χρήστης τον έχει ξεχάσει.

Η εικόνα 4.8 παρουσιάζει τη φόρμα εγγραφής νέου χρήστη στην πλατφόρμα. Ο χρήστης καλείται να συμπληρώσει προσωπικά στοιχεία όπως όνομα, επώνυμο, email και κωδικό πρόσβασης, καθώς και να επιλέξει τον ρόλο του (φοιτητής ή καθηγητής). Η διαδικασία εγγραφής έχει σχεδιαστεί με στόχο την απλότητα και την ταχεία ολοκλήρωση της καταχώρισης νέου λογαριασμού.

Εγγραφή

Το Επώνυμο σας

Το Όνομα σας

Email

Κωδικός

Εγγραφή

[Σύνδεση](#)

Εικόνα 4.8: Για Εγγραφή

Καλώς ήρθατε, admin@admin.com

Ενέργειες Διαχειριστή

Διαχείριση Μαθημάτων

Εικόνα 4.9: Αρχική - Admin

Στην εικόνα 4.9 εμφανίζεται η αρχική σελίδα μετά τη σύνδεση ως διαχειριστής (Admin). Ο διαχειριστής έχει πρόσβαση σε όλες τις βασικές λειτουργίες διαχείρισης του συστήματος, όπως η διαχείριση μαθημάτων και η παρακολούθηση των χρηστών.

Η εικόνα 4.10 απεικονίζει τη διαδικασία εισαγωγής νέου μαθήματος από τον διαχειριστή. Ο χρήστης συμπληρώνει τα απαιτούμενα πεδία, όπως το όνομα και την περιγραφή του μαθήματος.

Διαχείριση Μαθημάτων

Προσθήκη Νέου Μαθήματος

Όνομα Μαθήματος:

Περιγραφή:

Καθηγητής:

Προσθήκη

Υφιστάμενα Μαθήματα

 Διαγραφή

Εικόνα 4.10: Διαχείριση Μαθημάτων - Προσθήκη Νέου Μαθήματος 1 - Admin

Στην εικόνα 4.11 παρουσιάζεται η ολοκλήρωση της διαδικασίας προσθήκης νέου μαθήματος. Ο διαχειριστής επιβεβαιώνει την αποθήκευση των στοιχείων, με αποτέλεσμα το μάθημα να είναι πλέον διαθέσιμο στο σύστημα για περαιτέρω επεξεργασία και ανάθεση σε καθηγητές.

Διαχείριση Μαθημάτων

Προσθήκη Νέου Μαθήματος

Όνομα Μαθήματος:

Περιγραφή:

Καθηγητής:

Υφιστάμενα Μαθήματα

Μαθηματικά - Μαθηματικά Ι - 1ου Εξαμήνου

Δομημένος Προγραμματισμός - Δομημένος Προγραμματισμός

Εικόνα 4.11: Διαχείριση Μαθημάτων - Προσθήκη Νέου Μαθήματος 2 - Admin

Καλώς ήρθατε, mpakaros@gmail.com

Ενέργειες Καθηγητή

Εικόνα 4.12: Αρχική - Teacher

Η εικόνα 4.12 καταγράφει την αρχική οθόνη που εμφανίζεται στον καθηγητή μετά τη σύνδεση. Μέσα από αυτήν την οθόνη, ο καθηγητής έχει τη δυνατότητα να δει άμεσα τα μαθήματα στα οποία διδάσκει και να μεταβεί στις λειτουργίες διαχείρισης εργασιών και φοιτητών.

Τα Μαθήματά μου

Μαθηματικά Μαθηματικά Ι - 1ου Εξαμήνου	Διαχείριση Φοιτητών	Διαχείριση Εργασιών
Δομημένος Προγραμματισμός Δομημένος Προγραμματισμός	Διαχείριση Φοιτητών	Διαχείριση Εργασιών

Εικόνα 4.13: Τα Μαθήματά μου- Teacher

Στην εικόνα 4.13 εμφανίζεται η σελίδα «Τα Μαθήματά μου», όπου ο καθηγητής μπορεί να δει συγκεντρωτικά όλα τα μαθήματα που του έχουν ανατεθεί. Για κάθε μάθημα παρέχονται άμεσες επιλογές διαχείρισης φοιτητών και εργασιών, διευκολύνοντας τη γρήγορη πλοήγηση στις σχετικές λειτουργίες.

Φοιτητές Μαθήματος

Όνομα	Email	Ενέργειες
Kostas Nakidis	kostas@gmail.com	Αφαίρεση
Maria Nakou	maria@gmail.com	Αφαίρεση

Προσθήκη Φοιτητή στο Μάθημα

Επιλέξτε Φοιτητή:

[Προσθήκη](#)

Εικόνα 4.14: Διαχείριση Φοιτητών - Φοιτητές Μαθήματος - Teacher

Η εικόνα 4.14 παρουσιάζει τη σελίδα διαχείρισης φοιτητών για ένα επιλεγμένο μάθημα. Ο καθηγητής μπορεί να δει τους φοιτητές που είναι εγγεγραμμένοι στο μάθημα, να αφαιρέσει φοιτητές ή να προσθέσει νέους μέσω ειδικής φόρμας.

Εργασίες Μαθήματος

Προσθήκη Νέας Εργασίας

Νέα Εργασία 1 (2025-05-11 14:17:00 - 2025-05-11 19:18:00) Νέα Εργασία 1 Επεξεργασία Δες Υποβολές Διαγραφή
Εργασία 2 (2025-05-14 19:18:00 - 2025-05-20 19:18:00) Περιγραφή Εργασία2 Επεξεργασία Δες Υποβολές Διαγραφή
Εργασία 0 (2025-04-08 15:19:00 - 2025-05-05 15:19:00) Εργασία 0 Επεξεργασία Δες Υποβολές Διαγραφή

Εικόνα 4.15: Διαχείριση Εργασιών - Εργασίες Μαθήματος - Teacher

Όπως φαίνεται στην εικόνα 4.15, παρέχεται στον καθηγητή η δυνατότητα να δει και να διαχειριστεί τις εργασίες που έχουν ανατεθεί σε ένα συγκεκριμένο μάθημα. Η σελίδα αυτή περιλαμβάνει λειτουργίες για την προσθήκη νέων εργασιών, την επεξεργασία και τη διαγραφή των υπαρχουσών.

Νέα Εργασία

Τίτλος:

Επίλυση Ασκήσεων Παραγώγων

Περιγραφή:

Βιβλίο Σελ 44-46 - Ασκήσεις 1-20

Ημερομηνία Έναρξης:

05/15/2025 08:00 PM

Ημερομηνία Λήξης:

05/22/2025 11:55 PM

Αποθήκευση

Εικόνα 4.16: Νέα Εργασία - Teacher

Η εικόνα 4.16 απεικονίζει τη φόρμα δημιουργίας νέας εργασίας από τον καθηγητή. Στο σημείο αυτό, καθορίζονται στοιχεία όπως ο τίτλος της εργασίας, η περιγραφή και οι ημερομηνίες έναρξης και λήξης της προθεσμίας υποβολής.

Εργασίες Μαθήματος

Προσθήκη Νέας Εργασίας

Η εργασία προστέθηκε.

Νέα Εργασία 1 (2025-05-11 14:17:00 - 2025-05-11 19:18:00)

Νέα Εργασία 1

Επεξεργασία

Δες Υποβολές

Διαγραφή

Εργασία 2 (2025-05-14 19:18:00 - 2025-05-20 19:18:00)

Περιγραφή Εργασία2

Επεξεργασία

Δες Υποβολές

Διαγραφή

Εργασία 0 (2025-04-08 15:19:00 - 2025-05-05 15:19:00)

Εργασία 0

Επεξεργασία

Δες Υποβολές

Διαγραφή

Επίλυση Ασκήσεων Παραγώγων (2025-05-15 20:00:00 - 2025-05-22 23:55:00)

Βιβλίο Σελ 44-46 - Ασκήσεις 1-20

Επεξεργασία

Δες Υποβολές

Διαγραφή

Εικόνα 4.17: Μετά την προσθήκη Νέας Εργασίας - Teacher

Στην εικόνα 4.17 παρουσιάζεται η οθόνη επιβεβαίωσης μετά την επιτυχή προσθήκη νέας εργασίας. Η νέα εργασία εμφανίζεται πλέον στη λίστα των εργασιών του μαθήματος, διαθέσιμη προς ανάθεση και υποβολή από τους φοιτητές.

Επεξεργασία Εργασίας

Τίτλος:

Νέα Εργασία 1

Περιγραφή:

Νέα Εργασία 1

Ημερομηνία Έναρξης:

05/11/2025 02:17 PM



Ημερομηνία Λήξης:

05/11/2025 07:18 PM



Αποθήκευση Αλλαγών

Εικόνα 4.18: Επεξεργασία Εργασίας και ρύθμιση Deadlines - Teacher

Στην εικόνα 4.18 εμφανίζεται η σελίδα επεξεργασίας μιας υπάρχουσας εργασίας. Ο καθηγητής έχει τη δυνατότητα να τροποποιήσει τον τίτλο, την περιγραφή, καθώς και τις ημερομηνίες έναρξης και λήξης

της υποβολής. Η δυνατότητα αυτή εξασφαλίζει ευελιξία στον επαναπροσδιορισμό των χρονικών πλαισίων σύμφωνα με τις ανάγκες του μαθήματος.

Υποβολές Φοιτητών

Φοιτητής	Αρχείο	Ημ/νία Υποβολής	Βαθμός	Σχόλια	Ενέργειες
Kostas Nakidis	Δήψη	2025-05-11 11:21:55	7.00	-	<input type="text" value="Βαθμός"/> <input type="text" value="Σχόλια (προαιρετι"/> <input type="button" value="Βαθμολόγηση"/>

Εικόνα 4.19: Υποβολές αρχείων εργασιών των φοιτητών - Teacher

Η εικόνα 4.19 παρουσιάζει την οθόνη με τις υποβολές εργασιών από τους φοιτητές. Ο καθηγητής μπορεί να δει τα αρχεία που έχουν υποβληθεί, την ημερομηνία υποβολής και την κατάσταση της βαθμολόγησης, ενώ παρέχεται η δυνατότητα λήψης των αρχείων για αξιολόγηση.

Υποβολές Φοιτητών

Φοιτητής	Αρχείο	Ημ/νία Υποβολής	Βαθμός	Σχόλια	Ενέργειες
Kostas Nakidis	Δήψη	2025-05-11 11:21:55	7.00	-	<input type="text" value="Βαθμός"/> <input type="text" value="Σχόλια (προα"/> <input type="button" value="Βαθμολόγηση"/>
Maria Nakou	Δήψη	2025-05-11 16:14:31	Μη Βαθμολογημένο	-	<input type="text" value="Βαθμός"/> <input type="text" value="Σχόλια (προα"/> <input type="button" value="Βαθμολόγηση"/>

Εικόνα 4.20: Υποβολές εργασιών φοιτητών με μη βαθμολογημένη εργασία- Teacher

Όπως φαίνεται στην εικόνα 4.20, απεικονίζονται οι υποβολές φοιτητών που δεν έχουν ακόμη αξιολογηθεί. Η οθόνη αυτή παρέχει στον καθηγητή την ευχέρεια να προχωρήσει άμεσα στη βαθμολόγηση και στην καταχώρηση σχολίων για κάθε υποβολή.

Υποβολές Φοιτητών

Η εργασία βαθμολογήθηκε.

Φοιτητής	Αρχείο	Ημ/νία Υποβολής	Βαθμός	Σχόλια	Ενέργειες
Kostas Nakidis	Δήψη	2025-05-11 11:21:55	7.00	-	<input type="text" value="Βαθμός"/> <input type="text" value="Σχόλια (προαιρετ)"/> <input type="button" value="Βαθμολόγηση"/>
Maria Nakou	Δήψη	2025-05-11 16:14:31	10.00	Μπράβο	<input type="text" value="Βαθμός"/> <input type="text" value="Σχόλια (προαιρετ)"/> <input type="button" value="Βαθμολόγηση"/>

Εικόνα 4.21: Βαθμολογημένες Υποβολές εργασιών φοιτητών - Teacher

Η εικόνα 4.21 απεικονίζει τις εργασίες που έχουν ήδη αξιολογηθεί από τον καθηγητή. Για κάθε υποβολή εμφανίζεται η βαθμολογία και τα σχόλια που καταχωρήθηκαν, προσφέροντας μια σαφή εικόνα της διαδικασίας αξιολόγησης.

CourseWork System Αρχική maria@gmail.com (Student) Αποσύνδεση

Καλώς ήρθατε, maria@gmail.com

Ενέργειες Φοιτητή

Εικόνα 4.22: Αρχική σελίδα για τον φοιτητή – Student

Στην εικόνα 4.22 έχουμε την αρχική σελίδα που εμφανίζεται στον φοιτητή μετά τη σύνδεσή του στην πλατφόρμα. Μέσω της σελίδας αυτής, ο φοιτητής μπορεί να έχει άμεση πρόσβαση στα μαθήματα στα οποία είναι εγγεγραμμένος και να δει σχετικές ειδοποιήσεις για εργασίες και βαθμολογίες.

CourseWork System Αρχική maria@gmail.com (Student) Αποσύνδεση

Τα Μαθήματά μου

Μαθηματικά

Μαθηματικά I - 1ου Εξαμήνου

Εικόνα 4.23: Σελίδα με τα μαθήματα που συμμετέχει ο φοιτητής- Student

Η εικόνα 4.23 καταγράφει τη σελίδα όπου ο φοιτητής μπορεί να δει όλα τα μαθήματα στα οποία είναι εγγεγραμμένος. Παρέχεται η δυνατότητα γρήγορης πλοήγησης στο περιεχόμενο κάθε μαθήματος και στις διαθέσιμες εργασίες.

Εργασίες Μαθήματος

Νέα Εργασία 1 Νέα Εργασία 1 Υποβολή: 2025-05-11 14:17:00 έως 2025-05-11 19:18:00 Υποβολή Εργασίας
Εργασία 2 Περιγραφή Εργασία2 Υποβολή: 2025-05-14 19:18:00 έως 2025-05-20 19:18:00 Η υποβολή δεν έχει ξεκινήσει ακόμα.
Εργασία 0 Εργασία 0 Υποβολή: 2025-04-08 15:19:00 έως 2025-05-05 15:19:00 Η προθεσμία υποβολής έχει λήξει.
Επίλυση Ασκήσεων Παραγώγων Βιβλίο Σελ 44-46 - Ασκήσεις 1-20 Υποβολή: 2025-05-15 20:00:00 έως 2025-05-22 23:55:00 Η υποβολή δεν έχει ξεκινήσει ακόμα.

Εικόνα 4.24: Εργασίες για το Μάθημα Μαθηματικά – όπως το βλέπει ο φοιτητής- Εργασίες εντός και εκτός προθεσμίας - Student

Στην εικόνα 4.24 απεικονίζεται η σελίδα με τις εργασίες ενός μαθήματος, όπως εμφανίζονται στον φοιτητή. Οι εργασίες διαχωρίζονται οπτικά ανάλογα με το αν βρίσκονται εντός ή εκτός προθεσμίας, ενώ υποδεικνύεται επίσης αν έχει ήδη πραγματοποιηθεί υποβολή.

Υποβολή Εργασίας

Επιλέξτε Αρχείο:

Choose File No file chosen

[Υποβολή](#)

Εικόνα 4.25: Σελίδα για υποβολή αρχείου - Student

Η εικόνα 4.25 παρουσιάζει τη φόρμα υποβολής εργασίας από τον φοιτητή. Μέσω της συγκεκριμένης σελίδας, ο φοιτητής μπορεί να ανεβάσει το αρχείο της εργασίας του και να προχωρήσει στην επίσημη καταγραφή της υποβολής στο σύστημα.

Υποβολή Εργασίας

Η εργασία υποβλήθηκε με επιτυχία.

Επιλέξτε Αρχείο:

Choose File No file chosen

Υποβολή

Υφιστάμενη Υποβολή:

Αρχείο: 1746980071_ΑσκήσειςΜαθηματικών.pdf

Υποβλήθηκε: 2025-05-11 16:14:31

Εικόνα 4.26: Επιτυχή υποβολή αρχείου - Student

Στην εικόνα 4.26 φαίνεται το μήνυμα επιτυχούς ολοκλήρωσης της διαδικασίας υποβολής. Ο φοιτητής ενημερώνεται ότι η εργασία του έχει υποβληθεί με επιτυχία και καταγράφεται η σχετική χρονική στιγμή υποβολής.

Εργασίες Μαθήματος

Νέα Εργασία 1

Νέα Εργασία 1

Υποβολή: 2025-05-11 14:17:00 έως 2025-05-11 19:18:00

Έχετε ήδη υποβάλει εργασία.

Προβολή Υποβολής

Εργασία 2

Περιγραφή Εργασία2

Υποβολή: 2025-05-14 19:18:00 έως 2025-05-20 19:18:00

Η υποβολή δεν έχει ξεκινήσει ακόμα.

Εικόνα 4.27: Σελίδα με τις εργασίες – φαίνεται ότι έγινε η υποβολή - Student

Η εικόνα 4.27 απεικονίζει τη σελίδα όπου ο φοιτητής μπορεί να δει τις εργασίες του μαθήματος και να διαπιστώσει οπτικά ότι έχει ολοκληρώσει την υποβολή για συγκεκριμένες εργασίες, μέσα από σχετική ένδειξη στο περιβάλλον της πλατφόρμας.

Βαθμολογίες

Μάθημα	Εργασία	Βαθμός	Σχόλια
Μαθηματικά	Νεα Εργασία 1	Χωρίς Βαθμό	-

Εικόνα 4.28: Σελίδα με τις βαθμολογίες ανά εργασία – χωρίς βαθμό - Student

Η εικόνα 4.28 παρουσιάζει τη σελίδα όπου ο φοιτητής μπορεί να δει τις βαθμολογίες των υποβληθέντων εργασιών. Στην προκειμένη περίπτωση, δεν έχει ακόμη ολοκληρωθεί η αξιολόγηση από τον καθηγητή, και ως εκ τούτου εμφανίζεται σχετική ένδειξη ότι η εργασία παραμένει χωρίς βαθμό.

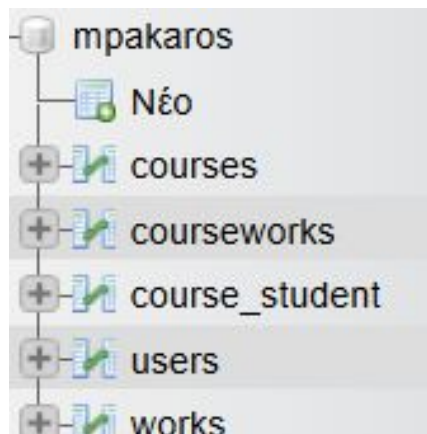
Βαθμολογίες

Μάθημα	Εργασία	Βαθμός	Σχόλια
Μαθηματικά	Νεα Εργασία 1	10.00	Μπράβο

Εικόνα 4.29: Σελίδα με τις βαθμολογίες ανά εργασία – με βαθμό - Student

Στην εικόνα 4.29 απεικονίζεται η σελίδα στην οποία ο φοιτητής έχει τη δυνατότητα να δει τις βαθμολογίες που έχουν αποδοθεί στις υποβληθείσες εργασίες του. Για κάθε εργασία εμφανίζεται η βαθμολογία που καταχωρήθηκε από τον καθηγητή, παρέχοντας στον φοιτητή εικόνα της επίδοσής του.

4.4 Η Βάση μας



Εικόνα 4.30: Οι πίνακες της βάσης μας

Η βάση δεδομένων της πλατφόρμας είναι σχεδιασμένη να καλύπτει πλήρως τις ανάγκες διαχείρισης χρηστών, μαθημάτων, εργασιών και υποβολών. Χρησιμοποιεί μια απλοποιημένη δομή, χωρίς σχέσεις Foreign Keys, με στόχο την ευκολία διαχείρισης και επέκτασης.

Πίνακες της Βάσης:

1. **users**

Περιέχει τα στοιχεία όλων των χρηστών της πλατφόρμας (φοιτητών, καθηγητών, διαχειριστών).

- id (PK)
- email
- password
- first_name
- last_name
- kind (0: Student, 1: Teacher, 2: Admin)
- created_at

2. **courses**

Περιέχει τις πληροφορίες των μαθημάτων.

- id (PK)
- name

- description
- teacher_id (συσχετίζεται λογικά με users.id)
- created_at

3. **course_student**

Διαχειρίζεται τις εγγραφές φοιτητών σε μαθήματα.

- id (PK)
- course_id
- student_id
- enrolled_at

4. **courseworks**

Καταγράφει τις εργασίες που έχουν ανατεθεί σε κάθε μάθημα.

- id (PK)
- course_id
- title
- description
- deadline_start
- deadline_end
- created_at

5. **works**

Περιέχει τις υποβολές των εργασιών από τους φοιτητές.

- id (PK)
- coursework_id
- student_id
- filename
- grade
- comments
- submitted_at
- graded_at

Περιγραφή Συσχετίσεων:

- Ένας **Καθηγητής** (users.kind = 1) μπορεί να έχει πολλά **Μαθήματα** (courses).
- Ένα **Μάθημα** μπορεί να έχει πολλές **Εργασίες** (courseworks).
- Ένας **Φοιτητής** μπορεί να εγγραφεί σε πολλά μαθήματα (course_student).
- Κάθε **Εργασία** (coursework) μπορεί να έχει πολλές **Υποβολές** (works) από διαφορετικούς φοιτητές.
- Η **υποβολή εργασίας** (works) σχετίζεται με έναν φοιτητή και μία εργασία.

4.5 Ασφάλεια συστήματος

Η ασφάλεια αποτελεί έναν από τους σημαντικότερους παράγοντες κατά τον σχεδιασμό και την ανάπτυξη της διαδικτυακής πλατφόρμας διαχείρισης φοιτητικών εργασιών. Η προστασία των προσωπικών δεδομένων των χρηστών, η διασφάλιση της ακεραιότητας των υποβολών και η αποτροπή μη εξουσιοδοτημένων ενεργειών κρίνονται απαραίτητες για την εύρυθμη λειτουργία της πλατφόρμας και τη συμμόρφωση με τις αρχές της ηθικής και νομικής προστασίας των δεδομένων.

Διαχείριση Πρόσβασης και Ρόλων

Η πλατφόρμα εφαρμόζει πολιτικές πρόσβασης βασισμένες σε ρόλους (Role-Based Access Control - RBAC). Κάθε χρήστης ανήκει σε έναν από τους εξής ρόλους:

- **Φοιτητής:** Πρόσβαση μόνο στα προσωπικά του δεδομένα, στα μαθήματα όπου είναι εγγεγραμμένος και στις εργασίες που του ανατίθενται. Μπορεί να υποβάλει εργασίες εντός των επιτρεπόμενων προθεσμιών και να δει τις βαθμολογίες του.
- **Καθηγητής:** Πρόσβαση σε όλα τα μαθήματα που του έχουν ανατεθεί, δυνατότητα διαχείρισης φοιτητών, δημιουργίας και αξιολόγησης εργασιών.
- **Διαχειριστής (Admin):** Απόλυτη πρόσβαση σε όλα τα δεδομένα και τις λειτουργίες της πλατφόρμας, συμπεριλαμβανομένης της διαχείρισης χρηστών και μαθημάτων.

Οι έλεγχοι πρόσβασης εφαρμόζονται τόσο σε επίπεδο διεπαφής χρήστη (Blade Templates) όσο και σε επίπεδο λογικής εφαρμογής (Controllers), διασφαλίζοντας ότι κάθε ενέργεια πραγματοποιείται μόνο από εξουσιοδοτημένους χρήστες.

Προστασία Υποβολών και Deadlines

Το σύστημα ενσωματώνει μηχανισμούς που εμποδίζουν την υποβολή εργασιών εκτός των προκαθορισμένων χρονικών παραθύρων. Η επεξεργασία ή η αντικατάσταση υποβολών μετά τη λήξη της προθεσμίας απαγορεύεται, ενώ οι φοιτητές μπορούν να δουν μόνο τις υποβολές τους χωρίς τη δυνατότητα τροποποίησης.

Επιπλέον ελέγχεται αν έχει ήδη πραγματοποιηθεί βαθμολόγηση μιας εργασίας. Σε αυτή την περίπτωση, οποιαδήποτε προσπάθεια νέας υποβολής απορρίπτεται, διασφαλίζοντας την ακεραιότητα της διαδικασίας αξιολόγησης.

Προστασία Δεδομένων Χρηστών και Αποθήκευσης Αρχείων

Τα προσωπικά δεδομένα των χρηστών (email, ονοματεπώνυμο, κωδικοί πρόσβασης) αποθηκεύονται στη βάση δεδομένων. Η αποθήκευση των κωδικών θα πρέπει να γίνεται με χρήση ισχυρών αλγορίθμων κρυπτογράφησης (π.χ. bcrypt), προκειμένου να διασφαλιστεί η μη αναγνώσιμη αποθήκευσή τους σε περίπτωση παραβίασης του συστήματος.

Τα αρχεία εργασιών αποθηκεύονται σε προστατευμένο φάκελο στο διακομιστή και η πρόσβαση σε αυτά επιτρέπεται μόνο μέσω ελεγχόμενων συνδέσμων και κατάλληλης επαλήθευσης του ρόλου του χρήστη.

Έλεγχος Καταχρηστικών Συμπεριφορών

Για την αποτροπή επιθέσεων τύπου brute force, spam εγγραφών ή μη εξουσιοδοτημένων προσπαθειών πρόσβασης, εφαρμόζονται τεχνικές όπως:

- Χρονικοί περιορισμοί μεταξύ διαδοχικών αποτυχημένων προσπαθειών σύνδεσης.
- Έλεγχος μοναδικότητας email και άλλων πεδίων κατά την εγγραφή.
- Χρήση Honeyrot πεδίων για την αποτροπή αυτοματοποιημένων εγγραφών από bots.
- Περιορισμός εγγραφών ανά IP σε συγκεκριμένο χρονικό διάστημα.

Αντιμετώπιση Ευπαθειών Ασφαλείας

Η πλατφόρμα προστατεύεται από κοινές ευπάθειες ασφαλείας μέσω:

- Επαλήθευσης και καθαρισμού εισόδου δεδομένων για αποτροπή SQL Injection και XSS επιθέσεων.

- CSRF Tokens σε όλες τις φόρμες υποβολής για την αποτροπή επιθέσεων Cross-Site Request Forgery.
- Κατάλληλης διαχείρισης Session ώστε να αποφεύγεται η υποκλοπή ταυτοτήτων χρηστών.
- Χρήσης HTTPS για την κρυπτογράφηση της επικοινωνίας μεταξύ client και server (σε παραγωγικά περιβάλλοντα).

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Παρουσιάστηκε και αναλύθηκε η διαδικτυακή πλατφόρμα διαχείρισης φοιτητικών εργασιών η οποία καλύπτει ανάγκες της σύγχρονης εκπαιδευτικής διαδικασίας όσον αφορά την υποβολή, αξιολόγηση και παρακολούθηση της προόδου των φοιτητών. Μέσα από την υλοποίηση βασικών λειτουργιών όπως η ηλεκτρονική διαχείριση μαθημάτων, η σωστή οριοθέτηση προθεσμιών υποβολής και η ασφαλής αξιολόγηση εργασιών, η πλατφόρμα προσφέρει ένα ολοκληρωμένο και αξιόπιστο περιβάλλον για την υποστήριξη του ακαδημαϊκού έργου. Παράλληλα δόθηκε ιδιαίτερη έμφαση στην ασφάλεια των δεδομένων και στη φιλικότητα του περιβάλλοντος χρήστη, συμβάλλοντας στην αποδοτικότερη διαχείριση των εκπαιδευτικών διαδικασιών και ενισχύοντας την αποτελεσματική επικοινωνία μεταξύ φοιτητών και διδασκόντων.

Η ανάπτυξη της διαδικτυακής πλατφόρμας διαχείρισης φοιτητικών εργασιών συνέβαλε ουσιαστικά στην απλοποίηση και αυτοματοποίηση κρίσιμων διαδικασιών τόσο για τους φοιτητές όσο και για τους καθηγητές. Η ηλεκτρονική διαχείριση των υποβολών, των μαθημάτων και των αξιολογήσεων εξάλειψε τις χρονοβόρες και συχνά αναποτελεσματικές παραδοσιακές μεθόδους, προσφέροντας μια πιο σύγχρονη και οργανωμένη προσέγγιση στην εκπαιδευτική διαδικασία. Το σύστημα ενσωματώνει αποτελεσματικούς μηχανισμούς ασφαλείας, προστατεύοντας τα προσωπικά δεδομένα των χρηστών και εξασφαλίζοντας την ακεραιότητα των υποβολών. Μέσα από την ορθή εφαρμογή ελέγχων πρόσβασης, διαχείρισης προθεσμιών και περιορισμών ρόλων, διασφαλίζεται ότι κάθε χρήστης έχει πρόσβαση μόνο στις πληροφορίες και λειτουργίες που του αναλογούν, περιορίζοντας τον κίνδυνο παραβίασης ή κατάχρησης των δεδομένων.

Η πλατφόρμα υποστηρίζει με επιτυχία τη διαχείριση διαφορετικών ρόλων χρηστών (φοιτητών, καθηγητών, διαχειριστών), επιτρέποντας την εύκολη προσαρμογή και επέκταση των λειτουργιών της. Η διακριτή οριοθέτηση των δικαιωμάτων κάθε ρόλου καθιστά το σύστημα ευέλικτο και προσαρμόσιμο σε διαφορετικά εκπαιδευτικά σενάρια, επιτρέποντας την εύκολη εισαγωγή νέων χρηστών και την ορθή διαχείριση των δραστηριοτήτων τους. Το σύστημα διευκολύνει σημαντικά τη διαδικασία αξιολόγησης των φοιτητικών εργασιών. Οι καθηγητές έχουν άμεση πρόσβαση στις υποβολές, μπορούν να παρέχουν σχόλια και να καταχωρούν βαθμολογίες, ενώ οι φοιτητές ενημερώνονται άμεσα για τα αποτελέσματα. Η ύπαρξη μηχανισμών για την καταγραφή σχολίων και την τεκμηρίωση της αξιολόγησης προσθέτει αξία στη διαδικασία, προάγοντας τη διαφάνεια και την εκπαιδευτική ανατροφοδότηση.

Η χρήση του Bootstrap και των Blade Templates εξασφαλίζει ένα φιλικό προς τον χρήστη και απόλυτα responsive περιβάλλον, προσβάσιμο από διαφορετικές συσκευές. Η απλή και καθαρή παρουσίαση των λειτουργιών επιτρέπει στους χρήστες να εκτελούν τις απαραίτητες ενέργειες χωρίς δυσκολία, καθιστώντας την πλατφόρμα ιδανική για χρήση ακόμα και από άτομα με περιορισμένη τεχνική κατάρτιση.

Να τονισθεί ότι για την συγγραφή της εργασίας χρησιμοποιήθηκαν πολύτιμα εργαλεία τεχνητής νοημοσύνης όπως είναι το chargpt και το gemini για την διόρθωση του συντακτικού και τη συμβολή σε διάφορα τμήματα τα οποία ερευνήθηκαν και υιοθετήθηκαν από τον συγγραφέα αφού στηρίχτηκαν σε πηγές.

Για την περαιτέρω ενίσχυση της ασφάλειας του συστήματος, προτείνεται η ενσωμάτωση πολυπαραγοντικής αυθεντικοποίησης κατά τη διαδικασία σύνδεσης. Με την ενεργοποίηση 2FA, οι χρήστες θα κληθούν να επιβεβαιώνουν την ταυτότητά τους μέσω ενός επιπλέον μηχανισμού (όπως κωδικός μέσω email ή SMS), προσφέροντας ισχυρότερη προστασία απέναντι σε επιθέσεις υποκλοπής κωδικών πρόσβασης.

Η προσθήκη λειτουργιών παραγωγής αναφορών και γραφημάτων που αποτυπώνουν την πορεία των φοιτητών και τα αποτελέσματα των αξιολογήσεων, θα προσφέρει σημαντική προστιθέμενη αξία στο σύστημα. Μέσω αυτών οι καθηγητές και οι διαχειριστές θα μπορούν να εντοπίζουν εκπαιδευτικές αδυναμίες, να παρακολουθούν τη συμμετοχή των φοιτητών και να λαμβάνουν πιο στοχευμένες αποφάσεις για τη βελτίωση της εκπαιδευτικής διαδικασίας.

Η ενσωμάτωση συστήματος ειδοποιήσεων μέσω email θα διευκολύνει σημαντικά την ενημέρωση των χρηστών σχετικά με κρίσιμες ενέργειες και προθεσμίες. Οι φοιτητές θα μπορούν να λαμβάνουν ειδοποιήσεις για νέες εργασίες, υπενθυμίσεις για επερχόμενες προθεσμίες και ενημερώσεις για τη βαθμολογία τους, ενώ οι καθηγητές θα ειδοποιούνται άμεσα για νέες υποβολές και εργασίες προς αξιολόγηση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://moodle.org/?lang=el>
- [2] <https://openedx.org/>
- [3] <https://edu.google.com/workspace-for-education/products/classroom/>
- [4] <https://www.php.net/manual/en/history.php.php>
- [5] <https://phptherightway.com/>
- [6] <https://www.w3schools.com/php/>
- [7] <https://www.zend.com/blog/php-8-and-beyond>
- [8] <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>
- [9] <https://laravel.com/docs>
- [10] <https://laravel-news.com/>
- [11] <https://webdesign.tutsplus.com/categories/laravel/courses>
- [12] <https://dolphinwebsolution.com/blog/advantages-and-disadvantages-of-laravel-development/>
- [13] <https://www.w3schools.com/MySQL/default.asp>
- [14] <https://www.indeed.com/career-advice/interviewing/interview-question-what-are-your-strengths-and-weaknesses>
- [15] <https://planetscale.com/blog/build-a-laravel-application-with-a-mysql-database>

ΠΑΡΑΡΤΗΜΑ Α

Κώδικες που χρησιμοποιήθηκαν.

Routes

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;
use App\Http\Controllers\StudentController;
use App\Http\Controllers\TeacherController;
use App\Http\Controllers\AdminController;

// -----
// Public Routes (Για όλους)
// -----

// Αρχική Σελίδα
Route::get('/', [UserController::class, 'welcome'])->name('welcome');

// Σελίδες Login/Logout/Register
Route::get('/login', [UserController::class, 'showLoginForm'])->name('login');
Route::post('/login', [UserController::class, 'login']);
Route::post('/logout', [UserController::class, 'logout'])->name('logout');
Route::get('/register', [UserController::class, 'showRegisterForm'])->name('register');
Route::post('/register', [UserController::class, 'register']);

// Dashboard μετά το login
Route::get('/dashboard', [UserController::class, 'dashboard'])->name('dashboard');

// -----
// Student Routes
// -----

// Προβολή των μαθημάτων του φοιτητή
Route::get('/student/courses', [StudentController::class, 'myCourses'])->name('student.courses');

// Υποβολή ή Τροποποίηση Εργασίας για συγκεκριμένο Coursework
Route::get('/student/upload/{courseworkId}', [StudentController::class, 'uploadWork'])->
>name('student.upload.form');
Route::post('/student/upload/{courseworkId}', [StudentController::class, 'uploadWork'])->
>name('student.upload.submit');

// Προβολή Βαθμών Φοιτητή
Route::get('/student/grades', [StudentController::class, 'viewGrades'])->name('student.grades');

// Προβολή Εργασιών (Courseworks) σε ένα μάθημα
```

```

Route::get('/student/courseworks/{courseId}', [StudentController::class, 'viewCourseworks']->name('student.courseworks'));

// -----
// Teacher Routes
// -----

// Προβολή των Μαθημάτων του Καθηγητή
Route::get('/teacher/courses', [TeacherController::class, 'listCourses']->name('teacher.courses'));

// Διαχείριση Φοιτητών στο Μάθημα
Route::get('/teacher/students/{courseId}', [TeacherController::class, 'viewStudents']->name('teacher.students'));
Route::post('/teacher/manage-students/{courseId}', [TeacherController::class, 'manageStudents']->name('teacher.manage.students'));

// Διαχείριση Courseworks (CRUD)
Route::get('/teacher/courseworks/{courseId}', [TeacherController::class, 'viewCourseworks']->name('teacher.courseworks'));
Route::get('/teacher/courseworks/{courseId}/create', [TeacherController::class, 'showCreateCoursework']->name('teacher.courseworks.create'));
Route::post('/teacher/courseworks/{courseId}/store', [TeacherController::class, 'storeCoursework']->name('teacher.courseworks.store'));
Route::get('/teacher/courseworks/edit/{courseworkId}', [TeacherController::class, 'showEditCoursework']->name('teacher.courseworks.edit'));
Route::post('/teacher/courseworks/update/{courseworkId}', [TeacherController::class, 'updateCoursework']->name('teacher.courseworks.update'));
Route::post('/teacher/courseworks/delete/{courseworkId}', [TeacherController::class, 'deleteCoursework']->name('teacher.courseworks.delete'));

// Προβολή Υποβολών Εργασιών από Φοιτητές για Συγκεκριμένο Coursework
Route::get('/teacher/courseworks/{courseworkId}/submissions', [TeacherController::class, 'viewSubmissions']->name('teacher.courseworks.submissions'));

// Βαθμολόγηση Υποβολής Εργασίας (Work)
Route::post('/teacher/grade-work/{workId}', [TeacherController::class, 'gradeWork']->name('teacher.grade.work'));

// -----
// Admin Routes
// -----

// Προβολή και Διαχείριση Μαθημάτων από τον Admin
Route::get('/admin/courses', [AdminController::class, 'listCourse']->name('admin.courses'));
Route::post('/admin/manage-course', [AdminController::class, 'manageCourse']->name('admin.manage.course'));

```

StudentController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;

class StudentController extends Controller
{
    public function viewCourseworks($courseId)
    {
        $studentId = Session::get('user_id');

        // Βρίσκουμε τα Courseworks του μαθήματος
        $courseworks = DB::table('courseworks')
            ->where('course_id', $courseId)
            ->get();

        // Ελέγχουμε αν ο φοιτητής έχει ήδη υποβάλει εργασία για κάθε coursework
        $submittedWorks = DB::table('works')
            ->where('student_id', $studentId)
            ->pluck('coursework_id')
            ->toArray();

        return view('student.courseworks.index', compact('courseworks', 'submittedWorks', 'courseId'));
    }

    // Προβολή μαθημάτων που είναι εγγεγραμμένος ο φοιτητής
    public function myCourses()
    {
        $studentId = Session::get('user_id');
        $courses = DB::table('course_student')
            ->join('courses', 'course_student.course_id', '=', 'courses.id')
            ->where('course_student.student_id', $studentId)
            ->get();

        return view('student.courses', compact('courses'));
    }
}
```

```

}

// Προβολή Courseworks (εργασιών) του μαθήματος και υποβολή/τροποποίηση υποβολής
public function uploadWork(Request $request, $courseworkId)
{
    $studentId = Session::get('user_id');

    // Φέρνουμε τα στοιχεία της εργασίας
    $coursework = DB::table('courseworks')->where('id', $courseworkId)->first();

    if (!$coursework) {
        return redirect()->back()->withErrors(['error' => 'Η εργασία δεν βρέθηκε.']);
    }

    $now = \Carbon\Carbon::now();
    $deadlineStart = \Carbon\Carbon::parse($coursework->deadline_start);
    $deadlineEnd = \Carbon\Carbon::parse($coursework->deadline_end);

    // Έλεγχος αν έχει ήδη υποβληθεί εργασία
    $existingWork = DB::table('works')
        ->where('student_id', $studentId)
        ->where('coursework_id', $courseworkId)
        ->first();

    // Αν είναι GET Request (Προβολή Φόρμας)
    if ($request->isMethod('get')) {
        if ($existingWork) {
            // Μπορεί μόνο να δει την υποβολή, όχι να υποβάλει ξανά
            return view('student.upload_work', [
                'work' => $existingWork,
                'courseworkId' => $courseworkId,
                'submissionAllowed' => false
            ]);
        }

        // Αν δεν έχει ξεκινήσει η περίοδος
        if ($now->lt($deadlineStart)) {
            return back()->withErrors(['error' => 'Η περίοδος υποβολής δεν έχει ξεκινήσει.']);
        }
    }
}

```

```

// Αν έχει λήξει η περίοδος
if ($now->gt($deadlineEnd)) {
    return back()->withErrors(['error' => 'Η περίοδος υποβολής έχει λήξει.']);
}

return view('student.upload_work', [
    'work' => null,
    'courseworkId' => $courseworkId,
    'submissionAllowed' => true
]);
}

// Αν είναι POST (Υποβολή Αρχείου)
if ($existingWork) {
    return back()->withErrors(['error' => 'Έχετε ήδη υποβάλει εργασία.']);
}

if ($now->lt($deadlineStart) || $now->gt($deadlineEnd)) {
    return back()->withErrors(['error' => 'Δεν επιτρέπεται η υποβολή αυτή τη στιγμή.']);
}

// Διαχείριση αρχείου
$file = $request->file('file');
if ($file) {
    $filename = time() . '_' . $file->getClientOriginalName();
    $file->move(public_path('uploads'), $filename);

    DB::table('works')->insert([
        'student_id' => $studentId,
        'coursework_id' => $courseworkId,
        'filename' => $filename,
        'submitted_at' => now(),
    ]);

    return back()->with('success', 'Η εργασία υποβλήθηκε με επιτυχία.');
```

```

}

return back()->withErrors(['error' => 'Παρακαλώ επιλέξτε αρχείο για υποβολή.']);
}

```

```

// Προβολή βαθμολογιών
public function viewGrades()
{
    $studentId = Session::get('user_id');
    $grades = DB::table('works')
        ->join('courseworks', 'works.coursework_id', '=', 'courseworks.id')
        ->join('courses', 'courseworks.course_id', '=', 'courses.id')
        ->where('works.student_id', $studentId)
        ->select('courses.name as course', 'courseworks.title as coursework', 'works.grade',
'works.comments')
        ->get();

    return view('student.grades', compact('grades'));
}
}

```

TeacherController

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;
use Illuminate\Support\Facades\Log;

class TeacherController extends Controller
{
    // Λίστα μαθημάτων του καθηγητή
    public function listCourses()
    {
        $teacherId = Session::get('user_id');
        $courses = DB::table('courses')
            ->where('teacher_id', $teacherId)
            ->get();

        return view('teacher.courses', compact('courses'));
    }
}

```

```

// Προβολή φοιτητών σε μάθημα
public function viewStudents($courseId)
{
    $students = DB::table('course_student')
        ->join('users', 'course_student.student_id', '=', 'users.id')
        ->where('course_student.course_id', $courseId)
        ->select('users.id', 'users.first_name', 'users.last_name', 'users.email')
        ->get();

    // Βρίσκουμε διαθέσιμους φοιτητές που ΔΕΝ είναι ήδη στο μάθημα
    $enrolledStudentIds = $students->pluck('id')->toArray();

    $availableStudents = DB::table('users')
        ->where('kind', 0) // 0 = Student
        ->whereNotIn('id', $enrolledStudentIds)
        ->get();

    return view('teacher.students', compact('students', 'courseId', 'availableStudents'));
}

// Διαχείριση φοιτητών στο μάθημα (προσθήκη/αφαίρεση)
public function manageStudents(Request $request, $courseId)
{
    if ($request->action == 'add') {
        DB::table('course_student')->insert([
            'course_id' => $courseId,
            'student_id' => $request->student_id,
            'enrolled_at' => now(),
        ]);
    } elseif ($request->action == 'remove') {
        DB::table('course_student')
            ->where('course_id', $courseId)
            ->where('student_id', $request->student_id)
            ->delete();
    }

    return back();
}

```

```

// Προσθήκη/Τροποποίηση Coursework (εργασίας μαθήματος)
public function manageCoursework(Request $request, $courseId)
{
    if ($request->action == 'create') {
        DB::table('courseworks')->insert([
            'course_id' => $courseId,
            'title' => $request->title,
            'description' => $request->description,
            'deadline_start' => $request->deadline_start,
            'deadline_end' => $request->deadline_end,
            'created_at' => now(),
        ]);
    } elseif ($request->action == 'update') {
        DB::table('courseworks')
            ->where('id', $request->coursework_id)
            ->update([
                'title' => $request->title,
                'description' => $request->description,
                'deadline_start' => $request->deadline_start,
                'deadline_end' => $request->deadline_end,
            ]);
    } elseif ($request->action == 'delete') {
        DB::table('courseworks')
            ->where('id', $request->coursework_id)
            ->delete();
    }

    return back();
}

// Βαθμολόγηση υποβολής εργασίας

public function gradeWork(Request $request, $workId)
{
    DB::table('works')
        ->where('id', $workId)
        ->update([
            'grade' => $request->grade,
            'comments' => $request->comments,
        ]);
}

```

```

        'graded_at' => now(),
    ]);

    return back()->with('success', 'Η εργασία βαθμολογήθηκε.');
```

```

}
```

```

// Προβολή όλων των Courseworks για ένα μάθημα
```

```

public function viewCourseworks($courseId)
```

```

{
```

```

    $courseworks = DB::table('courseworks')->where('course_id', $courseId)->get();
```

```

    return view('teacher.courseworks.index', compact('courseworks', 'courseId'));
}
```

```

// Προβολή φόρμας για δημιουργία Coursework
```

```

public function showCreateCoursework($courseId)
```

```

{
```

```

    return view('teacher.courseworks.create', compact('courseId'));
}
```

```

// Αποθήκευση νέου Coursework
```

```

public function storeCoursework(Request $request, $courseId)
```

```

{
```

```

    DB::table('courseworks')->insert([
```

```

        'course_id' => $courseId,
```

```

        'title' => $request->title,
```

```

        'description' => $request->description,
```

```

        'deadline_start' => $request->deadline_start,
```

```

        'deadline_end' => $request->deadline_end,
```

```

        'created_at' => now(),
    ]);
```

```

    return redirect()->route('teacher.courseworks', $courseId)->with('success', 'Η εργασία προστέθηκε.');
```

```

}
```

```

// Προβολή φόρμας επεξεργασίας Coursework
```

```

public function showEditCoursework($courseworkId)
```

```

{
```

```

    $coursework = DB::table('courseworks')->where('id', $courseworkId)->first();
```

```

    return view('teacher.courseworks.edit', compact('coursework'));
}
```

```

}
```

```

// Ενημέρωση Coursework
public function updateCoursework(Request $request, $courseworkId)
{
    DB::table('courseworks')->where('id', $courseworkId)->update([
        'title' => $request->title,
        'description' => $request->description,
        'deadline_start' => $request->deadline_start,
        'deadline_end' => $request->deadline_end,
    ]);

    return back()->with('success', 'Η εργασία ενημερώθηκε.');
```

```

}

// Διαγραφή Coursework
public function deleteCoursework(Request $request, $courseworkId)
{
    DB::table('courseworks')->where('id', $courseworkId)->delete();
    return back()->with('success', 'Η εργασία διαγράφηκε.');
```

```

}

// Προβολή Υποβολών για Συγκεκριμένο Coursework
public function viewSubmissions($courseworkId)
{
    $submissions = DB::table('works')
        ->join('users', 'works.student_id', '=', 'users.id')
        ->where('works.coursework_id', $courseworkId)
        ->select('works.*', 'users.first_name', 'users.last_name', 'users.email')
        ->get();

    return view('teacher.courseworks.submissions', compact('submissions', 'courseworkId'));
}
}

```

AdminController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class AdminController extends Controller
{
    // Προβολή μαθημάτων και καθηγητών
    public function listCourse()
    {
        $courses = DB::table('courses')->get();
        $teachers = DB::table('users')->where('kind', 1)->get();

        return view('admin.courses', compact('courses', 'teachers'));
    }

    // Διαχείριση μαθημάτων (προσθήκη/επεξεργασία/διαγραφή)
    public function manageCourse(Request $request)
    {
        if ($request->action == 'create') {
            DB::table('courses')->insert([
                'name' => $request->name,
                'description' => $request->description,
                'teacher_id' => $request->teacher_id,
                'created_at' => now(),
            ]);
        } elseif ($request->action == 'update') {
            DB::table('courses')
                ->where('id', $request->course_id)
                ->update([
                    'name' => $request->name,
                    'description' => $request->description,
                    'teacher_id' => $request->teacher_id,
                ]);
        } elseif ($request->action == 'delete') {
            DB::table('courses')
                ->where('id', $request->course_id)
```

```
        ->delete();  
    }  
  
    return back();  
}  
}
```